

**PROLATE SPHEROIDAL  
DIGITAL FILTERING**

**CENTRE FOR NEWFOUNDLAND STUDIES**

**TOTAL OF 10 PAGES ONLY  
MAY BE XEROXED**

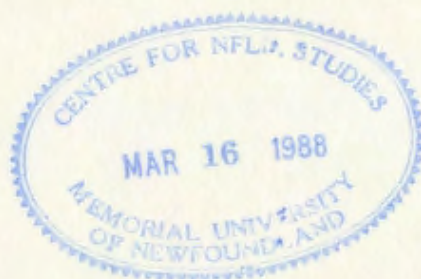
**(Without Author's Permission)**

**V. NEE LAKANTAN**





00010







# **PROLATE SPHEROIDAL DIGITAL FILTERING**

**BY**

***V. NEELAKANTAN, B.Sc. (Phys. ), B.E. (Elec. )***

© A thesis submitted to the School of Graduate  
Studies in partial fulfillment of the  
requirements for the degree of  
Master of Engineering

**Faculty of Engineering and Applied Science  
Memorial University of Newfoundland  
June 1986**

**St. John 's**

**Newfoundland**

**Canada**

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-33624-2



*To my parents*

*Varalakshmi & Visvanathan*

## ABSTRACT

The use of Digital Prolate Spheroidal Wave Functions (DPSWF) in the field of digital filtering has increased steadily during the past decade. The unique property: that DPSWF provide maximum concentration of signal energy in the passband of a low-pass filter has been the basis of most of the work done in the area of digital filter design involving the prolate functions. The maximization of energy corresponds to the largest eigenvalue, which belongs to the lowest order DPSWF, and the energy concentration decreases as the order increases.

It is, however, desirable to investigate the effect of higher order eigenfunctions in addition to the lowest order one on the filter characteristics and performance. This thesis is the outcome of such investigations. A suitably weighted linear combination of those DPSWF which are even functions of frequency is made to approximate an ideal low-pass characteristic in the minimum mean squared error (MMSE) sense.

Results employing various numbers of even prolate functions are obtained and compared with each other and with the maximum energy concentration case. It is shown that the flatness of the passband of a low-pass filter is greatly improved with the number of even prolate functions while maintaining a tolerable loss of energy in the passband.

One of the key features of the lowest order DPSWF, is that its side lobes are very low. This is a direct result of maximizing the concentration of energy within



a certain interval. This feature is very useful for harmonic analysis problems. This is demonstrated by applying the prolate function as a window in harmonic analysis using discrete Fourier transform. The performance of the prolate function is compared to that of many other popular window functions, in the present work.

## ACKNOWLEDGEMENTS

This thesis was completed at the Memorial University of Newfoundland, as a requirement towards the Master's degree, with support in the form of University Fellowship and graduate supplement, for which the author is deeply indebted.

The guidance from the supervisor, Dr. B.P. Sinha, Associate Professor at the Faculty of Engineering, has made this work possible and enjoyable. The author would like to express his appreciation and deep sense of gratitude towards his supervisor for many useful suggestions to improve the quality of this thesis. The University library facilities were the best the author has seen, and the Computing facilities were excellent in his opinion. The encouragement and inspiration shown by Dr. F.A. Aldrich, Dean of Graduate Studies, and Dr. G.R. Peters, Dean of Engineering, and the care rendered to the graduate students by the Associate Dean of Engineering, Dr. T.R. Chari, is highly appreciated.

The author would also like to thank Dr. P.P. Narayanaswami, Associate Professor at the Mathematics Department, for his introduction to the text-processing packages available in the UNIX operating system. Particularly, the documents written by Dr. P.P. Narayanaswami for processing equations and tables on the UNIX system, were very useful and concise, and the author owes a great deal of appreciation.

The author feels indebted to his parents for their encouragement. Thanks are due to Dr. A. Chockalingam, Assistant Professor at the Faculty of Medicine, and family, for their moral support which helped the author to overcome his



homesickness. Thanks are also due to the author's colleagues at the Faculty of Engineering, whose moral support made this thesis work less strenuous and the stay in St. John's more homely.

# TABLE OF CONTENTS

	Page
ABSTRACT	iii
ACKNOWLEDGEMENTS	v
LIST OF FIGURES	ix
CHAPTER 1 INTRODUCTION	1
1.1 Literature Review and Objective of the Thesis	1
1.2 Prolate Spheroidal Wave Functions	2
1.2.1 General Aspects	2
1.2.2 Computational Techniques	3
1.2.3 Relation to Band-limited & Time-limited Signals	6
1.2.4 The Discrete Case	8
1.3 Present Work	11
1.4 Thesis Organization	13
CHAPTER 2 THE DISCRETE PROLATE SPHEROIDAL FUNCTIONS	14
2.1 The Discrete Prolate Spheroidal Wave Functions (DPSWF)	14
2.2 The Discrete Prolate Spheroidal Sequences (DPSS)	17
2.3 Equations Connecting the DPSWF and the DPSS	18
2.4 Extremal Properties	20
2.4.1 Most Concentrated Band-limited Sequence	21
2.4.2 Index-limited Sequence with Most Concentrated Spectrum	23
2.5 The Discrete Prolate Filter	26
2.6 Computation of the DPSS & DPSWF	27
CHAPTER 3 FOURIER ADDITION OF THE DPSWF	30
3.1 Method for Improvement of Passband Performance	30
3.1.1 Least Squares Approximation and Fourier Series	30
3.1.2 Approximation Using the DPSWF	32
3.2 Results and Discussion	35
3.2.1 Filters with $N = 4$ and $N = 5$ , for $W = 0.20$	35
3.2.2 Filters with $N = 6$ and $N = 7$ , for $W = 0.20$	39
3.2.3 Filter with $N = 10$ and $W = 0.20$	42
3.2.4 Filter with $N = 10$ and $W = 0.15$	46
3.2.5 Comparison to the Chebyshev Filter	48
3.2.6 Summary of Comparisons	51



	Page
CHAPTER 4 PROLATE FUNCTIONS IN THE LIGHT OF HARMONIC ANALYSIS	53
4.1 Use of Windows in Filter Design	53
4.2 Use of Windows in Harmonic Analysis	56
4.2.1 The Discrete Fourier Transform	56
4.2.2 Spectral Leakage	60
4.3 Figures of Merit for Windows	62
4.3.1 Equivalent Noise Bandwidth	62
4.3.2 Processing Gain and Processing Loss	63
4.3.3 Scalloping Loss	65
4.3.4 Minimum Resolution Bandwidth	66
4.3.5 Summary and Other Merits	67
4.4 The Prolate Window	68
4.4.1 Comparison to Other Windows	68
4.4.2 Side Lobe Levels vs Worst Case Processing Loss	71
4.5 The Two Tone Detection Problem	73
4.5.1 The Tones at 5.0 and 10.0 Bins	76
4.5.2 The Tones at 5.5 and 10.0 Bins	78
4.5.3 The Tones at 5.25 and 10.0 Bins	80
CHAPTER 5 SUMMARY AND CONCLUSIONS	83
REFERENCES	86
APPENDIX PROGRAM LISTINGS	89

# LIST OF FIGURES

Fig. No.	Title	Page
1	$N = 4$ , $W = 0.20$ , Orders 0 & 2. Fourier Addition of the DPSWF.	36
2	$N = 5$ , $W = 0.20$ , Orders 0, 2 & 4. Fourier Addition of the DPSWF.	38
3	$N = 6$ , $W = 0.20$ , Orders 0, 2 & 4. Fourier Addition of the DPSWF.	40
4	$N = 7$ , $W = 0.20$ , Orders 0, 2, 4 & 6. Fourier Addition of the DPSWF.	41
5	$N = 10$ , $W = 0.20$ , Orders 0, 2 & 4. Fourier Addition of the DPSWF.	43
6	$N = 10$ , $W = 0.20$ , Orders 0, 2, 4 & 6. Fourier Addition of the DPSWF.	44
7	$N = 10$ , $W = 0.20$ , Orders 0, 2, 4, 6 & 8. Fourier Addition of the DPSWF.	45
8	$N = 10$ , $W = 0.15$ , Orders 0, 2 & 4. Fourier Addition of the DPSWF.	47
9	$N = 10$ , Order 0 ( $W = 0.20$ ), Orders 0, 2 & 4 ( $W = 0.15$ ). Fourier Addition of the DPSWF.	49
10	$N = 10$ , Orders 0, 2 & 4 ( $W = 0.15$ ). Comparison of Fourier Prolate & Chebyshev Filters.	50
11	$N = 10$ , Orders 0, 2 & 4 ( $W = 0.15$ ). Passband Performance of the Filters in Detail.	52
12	Comparison of Windows Based on Side Lobe Level and Worst Case Processing Loss.	72
13	Rectangle Window. Frequency Response of the Window.	74

Fig. No.	Title	Page
14	Blackman-Harris & Prolate Windows. Comparison of Frequency Response.	75
15	Input Tones at 5.0 and 10.0 Bins. Power Spectrum after Windowing.	77
16	Input Tones at 5.50 and 10.0 Bins. Power Spectrum after Windowing.	79
17	Input Tones at 5.25 and 10.0 Bins. Power Spectrum after Windowing.	81



# CHAPTER 1

## INTRODUCTION

### 1.1 Literature Review and Objective of the Thesis

Prolate spheroidal wave functions have extensive applications in many important fields such as electromagnetic and acoustic scattering, absorption of electromagnetic radiation by prolate spheroidal model of a human being, molecular physics etc. However, electromagnetic scattering by spheroids has been and continues to be one of the major areas of application of the spheroidal wave functions [1-6].

In addition to electromagnetic applications the prolate functions have also important applications in the fields of signal processing and digital filtering. This is due to the inherent band-limiting characteristics of these functions, as shown by Slepian et. al. [7] in 1961. The series of papers on 'Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty', by D. Slepian, H.O. Pollak and H.J. Landau [7-11], covered many aspects, including discrete ones, of the prolate functions, as applied to signal theory. They discussed the prolate functions in the light of the Fourier transform. The discrete versions of these functions, the so called discrete prolate functions, were applied to the field of digital filtering by Tufts and Francis [12] in 1970, and independently by Papoulis and Bertran [13] in 1972, prior to Slepian's paper [11] on the discrete version of the prolate spheroidal wave functions.

Eberhard [14] used the discrete functions in the calculation of power spectra, in 1973. Jain and Ranganath [15], used them in an extrapolation algorithm for discrete signals, in 1981. Dickey, Shanmugam and Green [16, 17] used the prolate function  $\psi_1(c, x)$  (in the notation of Slepian [7], this is the prolate function of order 1) for edge detection in picture processing, in 1977, 79. Durrani and Chapman [18, 19] designed IIR (Infinite Impulse Response) digital filters using the discrete prolate functions, in 1981, 84.

In this thesis, the effect of higher order discrete prolate spheroidal eigenfunctions, in addition to the lowest order eigenfunction, on the characteristics of a low-pass digital filter, is investigated. Also the eigenfunction of the lowest order is applied as a window, in analyzing the harmonic contents of a digital input sequence using discrete Fourier transform. The performance of this window is compared with that of other popular windows.

## 1.2 Prolate Spheroidal Wave Functions

In this section a brief account of prolate spheroidal functions, their computations and their relation to signal processing and digital filtering is given.

### 1.2.1 General Aspects

The prolate spheroidal wave functions are the solutions of the differential equation [20, 7],

$$(1 - t^2) \frac{d^2 u}{dt^2} - 2t \frac{du}{dt} + \left( \lambda - c^2 t^2 - \frac{m^2}{1 - t^2} \right) u = 0, \quad (1.1)$$

where  $c$  is real,  $\lambda$  and  $m$  are separation constants and  $\lambda$  can take only



certain (countably infinite number of) discrete values. For other values of  $\lambda$ , series solutions to (1.1) do not converge. For single-valuedness of the prolate functions,  $m$  has to be an integer, and it is customary to assume that  $m$  is either zero or positive. Since  $\lambda$  can take only certain discrete values for each  $m$ , it is usual to designate it by  $\lambda_{mn}$ , where  $m, = 0, 1, 2, \dots$  and  $n \geq m$ . For each  $\lambda_{mn}$ , there is a unique solution  $S_{mn}(c, t)$  to the above equation, called the angular prolate spheroidal functions. For each  $\lambda_{mn}$ , a second set of solutions exists. These are called the radial prolate spheroidal functions of the first kind,  $R_{mn}^{(1)}(c, t)$ . These radial functions are simply scaled versions of the angular functions.

Many authors [20-23], have dealt with these functions in great detail. However, there are diversities in the notations and the normalization schemes of these functions, in literature. The various notations in vogue are mentioned in [20, p.14]. In this thesis, Flammer's notations [20] are used as already adapted by Slepian, et al. [7-10].

### 1.2.2 Computational Techniques

The computation of the spheroidal wave functions is quite complex. A brief survey of the computational techniques for these functions, is given here. The functions are estimated by means of series expansions using other known functions like the Legendre functions, spherical Bessel, Neumann and Hankel functions. The expansion coefficients obey a recursion relation that has three (or more) terms. Convergent solutions to these recursion formulae exist, only for



certain discrete  $\lambda$  values in (1.1), as stated before. For  $c = 0$ , (1.1) reduces to the differential equation that is satisfied by the associated Legendre functions. For  $c \neq 0$ , the angle functions are expanded as an infinite series in the Legendre functions, [20]

$$S_{mn}(c, t) = \sum_{r=0,1}^{\infty}{}' d_r^{mn}(c) P_{m+r}^m(t). \quad (1.2)$$

The prime over the summation sign indicates that the summation is over even indices whenever  $n - m$  is even, and is over odd indices whenever  $n - m$  is odd. The coefficients  $d_r^{mn}(c)$ , satisfy the recursion relation,

$$\begin{aligned} & \frac{(2m+r+2)(2m+r+1)c^2}{(2m+2r+3)(2m+2r+5)} d_{r+2}^{mn}(c) \\ & + \left[ (m+r)(m+r+1) - \lambda_{mn}(c) + \frac{2(m+r)(m+r+1) - 2m^2 - 1}{(2m+2r-1)(2m+2r+3)} c^2 \right] d_r^{mn}(c) \\ & + \frac{r(r-1)c^2}{(2m+2r-3)(2m+2r-1)} d_{r-2}^{mn}(c) = 0, \quad (r \geq 0) \end{aligned} \quad (1.3)$$

This recursion relation is a linear homogeneous second-order difference equation, and has two independent, non-trivial solutions. Only one of them converges as  $r$  approaches  $\infty$ , and this solution is used in expanding the angle functions, as in (1.2). For small values of  $c^2$ , the eigenvalues  $\lambda_{mn}(c)$  and the expansion coefficients  $d_r^{mn}(c)$  can be obtained, as a power series in  $c^2$ . But this method is not very accurate. Bouwkamp [20, p.20] suggested a correction scheme for the eigenvalues and this improved accuracy from two to six decimal places.

An alternate approach, was to expand the angle functions  $S_{mn}(c, t)$  as a power series in  $(1 - t^2)$ , and this leads to the following relations [20, p.23]

$$S_{mn}(c, t) = (1 - t^2)^{\frac{m}{2}} \sum_{k=0}^{\infty} c_{2k}^{mn} (1 - t^2)^k, \quad (n-m) \text{ even} \quad (1.4a)$$

$$S_{mn}(c, t) = t (1 - t^2)^{\frac{m}{2}} \sum_{k=0}^{\infty} c_{2k}^{mn} (1 - t^2)^k, \quad (n-m) \text{ odd} \quad (1.4b)$$

The expansion coefficients  $c_{2k}^{mn}$  are related to the coefficients  $d_{2r}^{mn}$ , seen earlier, by the equations

$$c_{2k}^{mn} = \frac{1}{2^m k! (m+k)!} \sum_{r=k}^{\infty} \frac{(2m+2r)!}{(2r)!} (-r)_k \left( m+r+\frac{1}{2} \right)_k d_{2r}^{mn}, \quad (n-m) \text{ even} \quad (1.5a)$$

$$c_{2k}^{mn} = \frac{1}{2^m k! (m+k)!} \sum_{r=k}^{\infty} \frac{(2m+2r+1)!}{(2r+1)!} (-r)_k \left( m+r+\frac{3}{2} \right)_k d_{2r}^{mn}, \quad (n-m) \text{ odd} \quad (1.5b)$$

The subscript  $k$  in  $(r)_k$  implies that

$$(r)_k = r(r+1) \dots (r+k-1). \quad (1.6)$$

This method of expansion is useful for computing the radial functions.

Other expansion functions have also been attempted. However, the Legendre functions are widely used for the angle function computation. Radial prolate spheroidal wave functions of the first kind,  $R_{mn}^{(1)}(c, t)$ , are expressed as a series of spherical Bessel functions which converges rapidly. A third set of solutions for the differential equation in (1.1) is  $R_{mn}^{(2)}(c, t)$ , the radial prolate spheroidal wave

functions of the second kind. These are computed using a series expansion involving the spherical Neumann functions. The computations of  $R_{mn}^{(2)}(c, t)$ , was beset with convergence problems [20, p.32]. This problem was overcome by a method of integration proposed by Sinha & MacPhie [24]. This method reduced computation time on the computer considerably and also improved the accuracy of the eigenvalue to several decimal places.

### 1.2.3 Relation to Band-limited & Time-limited Signals

It is well known that a signal cannot be time-limited and band-limited at the same time. While strict limiting is not possible, one might try to see how much time-limited a band-limited signal can be or how much band-limited a finite duration signal can be. The 'amount' of time-limiting (or band-limiting) can be measured in terms of the concentration of signal energy within the intervals of interest. The solution to the problem of maximizing these concentrations is the prolate spheroidal wave function of the lowest order and degree. The problem has been treated well in the papers by Slepian et al. [7-10]. In the following paragraphs, their work is summarized.

Functions  $f(t)$ , that are defined for all time  $t$ , and are square integrable in the absolute, over the entire range of  $t$  on the real axis (from  $-\infty$  to  $\infty$ ), are said to belong to a class of functions denoted by  $L_{\infty}^2$  [7]. The energy of a signal, denoted by  $E$ , is given by the relation

$$E = \int_{-\infty}^{\infty} |f(t)|^2 dt. \quad (1.7)$$



The functions  $f(t)$ , have associated with them, a Fourier transform  $F(\omega)$  defined as

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt. \quad (1.8)$$

Here  $\omega$  denotes angular frequency and  $\omega/2\pi$  is the frequency variable  $f$ . Functions  $f(t)$ , whose Fourier transform vanishes outside a certain interval  $-\Omega < \omega < \Omega$  are said to be band-limited (and of bandwidth  $2\Omega$ ), and belong to the class of band-limited functions  $\mathbf{B}$ , which is thus a subset of  $\mathbf{L}_{\infty}^2$ . Functions in  $\mathbf{B}$  can be written as a finite inverse Fourier transform,

$$f(t) = \frac{1}{2\pi} \int_{-\Omega}^{\Omega} F(\omega) e^{j\omega t} d\omega. \quad (1.9)$$

On the other hand there are functions in  $\mathbf{L}_{\infty}^2$  that vanish outside a certain time interval  $-T/2 < t < T/2$ . Such functions are called time-limited functions and constitute a class denoted as  $\mathbf{D}$ , which is also a subset of  $\mathbf{L}_{\infty}^2$ . Given any function  $f(t)$  in  $\mathbf{L}_{\infty}^2$  one can obtain a band-limited or time-limited version of  $f(t)$ , by the operators  $B$  and  $D$  respectively, given by the following equations,

$$B : Bf(t) = \frac{1}{2\pi} \int_{-\Omega}^{\Omega} F(\omega) e^{j\omega t} d\omega, \quad (1.10)$$

$$D : Df(t) = \begin{cases} f(t) & |t| \leq T/2 \\ 0 & |t| > 0. \end{cases} \quad (1.11)$$

A function in  $\mathbf{B}$ , that has maximum energy concentration in the interval

$(-T/2, T/2)$  is the angular prolate spheroidal wave function  $S_{00}(c, t)$ . Incidentally,  $c$  here refers to the time-bandwidth product  $\Omega T/2$ . A function in  $L_{\infty}^2$ , when first time-limited and then band-limited, becomes a member of  $B$  with smaller total energy. The function that loses the smallest fraction of energy, in this transformation, is once again  $S_{00}(c, t)$  (except for a scalar factor). The ratio of the total energy of the transformed function in  $B$ , to the original function in  $L_{\infty}^2$  is  $\lambda_{00}$ , the eigenvalue corresponding to  $S_{00}(c, t)$ . Let the above transformation, viz., first time-limiting and then band-limiting, be performed on a function which is a member of  $B$  (instead of  $L_{\infty}^2$ , as assumed before). Again it is  $S_{00}(c, t)$ , that loses the smallest fraction of total energy under such a transformation. This time, the ratio of total energy of the transformed function to that of the original one is  $\lambda_{00}^2$ . Thus the prolate spheroidal wave function  $S_{00}(c, t)$ , has several useful extremal properties with respect to the band-limiting and time-limiting operations. That, this provides the basis for the designing of finite length digital filters, will be treated in later chapters.

#### 1.2.4 The Discrete Case [11]

Associated with the angle functions  $S_{0n}(c, t)$  are certain scaled, band-limited functions  $\psi_n(c, t)$  defined by

$$\psi_n(c, t) = \frac{\sqrt{\lambda_{0n}(c)}}{u_n(c)} S_{0n}(c, 2t/T) \quad (1.12)$$

where

$$[u_n(c)]^2 = \int_{-1}^1 [S_{0n}(c, t)]^2 dt, \quad (1.13)$$

and  $[u_n(c)]^2$  is thus the energy of  $S_{0n}(c, t)$  in the interval  $(-1, 1)$ . These scaled functions  $\psi_n(c, t)$  satisfy the integral equation

$$\lambda_{0i}(c) \psi_i(c, t) = \int_{-T/2}^{T/2} \frac{\sin \Omega(t-s)}{\pi(t-s)} \psi_i(c, s) ds, \quad i = 0, 1, 2, \dots \quad (1.14)$$

The kernel of this equation is easily identified as the sinc function, scaled in magnitude. The digitized version of the above equation is [11, p.1376],

$$\lambda_k(N, W) v_n^{(k)}(N, W) = \sum_{m=0}^{N-1} \frac{\sin 2\pi W(n-m)}{\pi(n-m)} v_m^{(k)}(N, W), \quad (1.15)$$

$$n = 0, \pm 1, \pm 2, \dots$$

where  $\{v_n^{(k)}(N, W)\}$  are called the Discrete Prolate Spheroidal Sequences (DPSS), for each  $k = 0, 1, 2, \dots, N-1$ . In the index set  $(0, N-1)$ , the DPSS obeys the following difference equation

$$\begin{aligned} \frac{1}{2} n (N-n) v_{n-1}^{(k)}(N, W) + \left[ \cos 2\pi W \left( \frac{N-1}{2} - n \right)^2 - \theta_k(N, W) \right] v_n^{(k)}(N, W) \\ + \frac{1}{2} (n+1) [N-1-n] v_{n+1}^{(k)}(N, W) = 0, \\ k, n = 0, 1, \dots, N-1. \end{aligned} \quad (1.16)$$

(The significance of these equations, will be discussed in the next chapter). It should be noted that, in the continuous case, the prolate spheroidal functions satisfy a second-order differential equation, and in the discrete case they satisfy a second-order difference equation.



The DPSS, in the index set  $(0, N-1)$ , has an associated spectrum defined in the following equation

$$U_k(N, W; f) = \epsilon_k \sum_{n=0}^{N-1} v_n^{(k)}(N, W) e^{-j\pi(N-1-2n)f}$$

$$k = 0, 1, \dots, N-1, \quad (1.17)$$

where  $\epsilon_k$  is a constant ( $\epsilon_k = 1$  for even  $k$  and  $\epsilon_k = j$  for odd  $k$ ). These functions,  $U_k(N, W; f)$  are called the Discrete Prolate Spheroidal Wave Functions (DPSWF), and are continuous periodic functions of the frequency variable  $f$ . In all the above equations,  $N$  is called the length of the sequence, and  $W$  is the normalized bandwidth. By the very process of sampling of a continuous-time signal (such as  $\psi_n(c, t)$  in our case), the frequency spectrum extends to the entire frequency axis. Consequently the notion of bandwidth is limited to the fundamental interval  $(-1/2, 1/2)$ , the Nyquist interval for unit sampling period. Hence  $|W| \leq 1/2$  in the above expressions.

A sequence that is band-limited to  $-W \leq f \leq W$  (its spectrum vanishes outside this interval), most concentrated on the index set  $(0, N-1)$  is the DPSS  $\{v_n^{(0)}(N, W)\}$ . As before, this is true to within a scalar constant factor. An index-limited sequence with spectrum most concentrated in  $-W \leq f \leq W$ , is again the DPSS  $\{v_n^{(0)}(N, W)\}$  on the index set  $(0, N-1)$ . The spectrum for this sequence, as seen before,  $U_0(N, W; f) \exp(j\pi(N-1)f)$ , which corresponds to the DPSWF of the lowest order and is the function that is considered in detail in the present work. The concentration of the signal is, as before, measured in terms of the

energy defined in

$$E \triangleq \sum_{n=-\infty}^{\infty} |v_n^{(k)}(N, W)|^2 = \int_{-1/2}^{1/2} |U_k(N, W; f)|^2 df \quad (1.18)$$

$$k = 0, 1, \dots, N-1.$$

The energy concentration for both these extremal cases is  $\lambda_0(N, W)$ . Thus the discrete prolate spheroidal sequence  $\{v_n^{(0)}(N, W)\}$  is very similar to its continuous counterpart  $S_{00}(c, t)$ , in its extremal properties.

### 1.3 Present Work

In the terminology of digital filters, the DPSS  $\{v_n^{(0)}(N, W)\}$  for  $n = 0, 1, \dots, N-1$ , would be called the filter coefficients for a Finite Impulse Response (FIR) filter. The frequency response of this digital filter is thus related to the DPSWF  $U_0(N, W; f)$ . Since  $U_0(N, W; f)$  has its energy concentrated in the interval  $(-W, W)$ , it has a low-pass characteristics. The low-pass filter it represents, shall henceforth be referred to as the discrete prolate filter or prolate filter for brevity.

Since the criterion is one of maximum energy concentration in the passband  $[(-W, W)]$  interval, the prolate filter has very low side lobe levels. For a filter of order  $N = 4$  and normalized bandwidth  $W = 0.20$ , the highest side lobe is about  $-20$  dB (this is with respect to the gain at d.c. which is the maximum and thus  $0$  dB). It is customary to represent the frequency response on a log scale, and to normalize it with respect to the gain at zero frequency. This side lobe level is a good one for a filter of that small length. As the filter order  $N$ , or the



bandwidth  $W$  is increased, the side lobe levels drop further down. For example, at  $N = 10$ ,  $W = 0.20$  the filter has a highest side lobe at a level of  $-50$  dB. The performance in the interval  $(-W, W)$ , called the passband characteristic is, however, not suitable for many filtering applications. The reason for this is a sloping response curve in the passband. To get an idea of how sloping the curve can be, for the case  $N = 4$ ,  $W = 0.20$ , at the edge of the passband the gain falls to  $-30$  dB. Between the zero frequency point and the passband edge, the response curve slopes smoothly. This limits the application of the prolate filter to the area of spectral analysis. The higher order prolate functions have lesser energy concentrations in their passband, but the passband sloping is also lesser. The motivation for the present work is to make use of these higher order functions to improve passband flatness and examine it in the light of Chebyshev filter, which is well known for its flat passband characteristics. The method to achieve this and the results obtained are given in Chapter 3.

The frequency response of the prolate filter based on the discrete prolate function of the lowest order, seems well suited for applications in harmonic analysis of data samples. This is tested in the present work, wherein the DPSS  $\{v_n^{(0)}(N, W)\}$  on the index set  $(0, N-1)$ , is construed as a data window, and it is used to analyze, with discrete Fourier transform, a finite set of samples for harmonic contents. Its performance is compared to that of Blackman-Harris window. Also, a few important figures of merit are computed for the prolate window, and it is compared to other familiar windows based on these figures of merit.



## 1.4 Thesis Organization

In the following chapter, the important properties of the DPSWF and the DPSS, and the methods to compute these prolate functions, are given. In Chapter 3, the method to improve flatness is discussed and the results of applying the method are presented. Chapter 4 contains the use of the discrete prolate functions in harmonic analysis problems. Several popular data windows are compared to the prolate sequence (DPSS  $\{v_n^{(0)}(N, W)\}$ ), that is used as a data window. The figures of merit for these windows, as defined in [25], are computed and compared. The problem of two-tone detection, presented in [25], is attempted with the prolate window (so named for brevity, since it uses the DPSS), and the results are presented in this chapter. Chapter 5 gives the conclusions drawn from this study. The computer programs are listed in the appendix.

## CHAPTER 2

### THE DISCRETE PROLATE SPHEROIDAL FUNCTIONS

In this chapter, the equations defining the DPSWF and the DPSS, and their interconnection, are presented. Some extremal properties of these discrete prolate functions are then presented. The computational methods employed in evaluating these functions are provided in the final section of this chapter.

#### 2.1 The Discrete Prolate Spheroidal Wave Functions [11]

The discrete prolate spheroidal wave functions (DPSWF),  $U_k(N, W; f)$  and their associated eigenvalues  $\lambda_k(N, W)$  evolve as the real solution to the integral equation,

$$\int_{-W}^W \frac{\sin N \pi (f - f')}{\sin \pi (f - f')} U_k(N, W; f') df' = \lambda_k(N, W) U_k(N, W; f) \quad (2.1)$$

where  $-\infty < f < \infty$ ,  $k = 0, 1, \dots, N-1$ . The DPSWF are normalized so that

$$\int_{-1/2}^{1/2} |U_k(N, W; f)|^2 df = 1, \quad (2.2a)$$

$$U_k(N, W; f) \Big|_{f=0} \geq 0 \quad \frac{d}{df} [U_k(N, W; f)] \Big|_{f=0} \geq 0. \quad (2.2b)$$

The eigenvalues are real, distinct and positive and are arranged in a descending order

$$\lambda_0(N, W) > \lambda_1(N, W) > \lambda_2(N, W) \dots > \lambda_{N-1}(N, W) > 0. \quad (2.3)$$

Since the kernel of the above integral equation can be written as

$$\frac{\sin N \pi (f - f')}{\sin \pi (f - f')} = \sum_{k=-(N-1)/2}^{(N-1)/2} e^{jk 2\pi f} e^{-jk 2\pi f'}, \quad (2.4)$$

it is degenerate [26]. Hence the integral equation has only  $N$  non-zero eigenvalues, and so  $k$  takes on the values in the range  $(0, N-1)$ .

The DPSWF are orthogonal in two different intervals. They are orthogonal in the  $(-W, W)$  interval, i.e.,

$$\int_{-W}^W U_i(N, W; f) U_j(N, W; f) df = \lambda_i \delta_{ij} \quad (2.5)$$

and they are orthonormal in the  $(-1/2, 1/2)$  interval, i.e.,

$$\int_{-1/2}^{1/2} U_i(N, W; f) U_j(N, W; f) df = \delta_{ij}. \quad (2.6)$$

In the above expressions  $\delta_{ij}$  is the Kronecker delta, i.e.,

$$\delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases} \quad i, j = 0, 1, \dots, N-1. \quad (2.7)$$

The DPSWF  $U_k(N, W; f)$ , are periodic functions of  $f$ . It has a period of 1 if  $N$  is odd, and a period of 2 if  $N$  is even. It is also symmetric about the origin, i.e. about the  $f = 0$  axis. Thus, it obeys the following symmetry relation

$$U_k(N, W; f+1) = (-1)^{N-1} U_k(N, W; f). \quad (2.8)$$



The DPSWF  $U_k(N, W; f)$ , has  $k$  zeros in the interval  $(-W, W)$  and  $N-1$  zeros in  $(-1/2, 1/2)$ . It is an even function about the origin whenever  $k$  is even, and is odd whenever  $k$  is odd.

These functions can also be found as the solution to the differential equation

$$\begin{aligned} & \frac{d}{d\omega} [\cos \omega - \cos 2\pi W] \frac{dU_k(N, W; f)}{d\omega} \\ & + \left[ \frac{1}{4} (N^2 - 1) \cos \omega - \theta_k(N, W) \right] U_k(N, W; f) = 0 \end{aligned} \quad (2.9)$$

where  $\omega$  is the angular frequency, and  $\omega = 2\pi f$ . The  $\theta_k(N, W)$ 's for  $k = 0$  to  $N-1$ , are the eigenvalues of the tridiagonal matrix  $\sigma(N, W)$  whose elements are given by,

$$\sigma(N, W)_{ij} = \begin{cases} \frac{1}{2} i (N-1), & j = i-1 \\ \left[ \left( \frac{N-1}{2} \right) - i \right]^2 \cos 2\pi W, & j = i \\ \frac{1}{2} (i+1) (N-1-i), & j = i+1 \\ 0, & |j-i| > 1. \end{cases} \quad (2.10)$$

$i, j = 0, 1, \dots, N-1.$

The  $\theta_k(N, W)$ 's are real and distinct and are arranged in a descending order, so that

$$\theta_0(N, W) > \theta_1(N, W) > \dots > \theta_{N-1}(N, W), \quad (2.11)$$

like the  $\lambda_k(N, W)$ 's in (2.3).

## 2.2 The Discrete Prolate Spheroidal Sequences [11]

The DPSS  $\{v_n^{(k)}(N, W)\}$  is the solution to the system of equations

$$\sum_{m=0}^{N-1} \frac{\sin 2\pi W(n-m)}{\pi(n-m)} v_m^{(k)}(N, W) = \lambda_k(N, W) v_n^{(k)}(N, W), \quad (2.12)$$

$$n = 0, \pm 1, \pm 2, \dots$$

for each  $k = 0, 1, \dots, N-1$ .

These sequences are normalized to unit energy so that

$$\sum_{j=0}^{N-1} [v_j^{(k)}(N, W)]^2 = 1, \quad (2.13)$$

and are arranged so that

$$\sum_{j=0}^{N-1} v_j^{(k)}(N, W) \geq 0, \quad \sum_{j=0}^{N-1} (N-1-2j) v_j^{(k)}(N, W) \geq 0. \quad (2.14)$$

The eigenvalues  $\lambda_k(N, W)$  associated with the DPSS in (2.12), are the same, as the eigenvalues of the DPSWF, in (2.1). These eigenvalues can be computed as the eigenvalues of the  $N \times N$  matrix  $\rho(N, W)$  with element in the  $m$ th row,  $n$ th column given by

$$\rho(N, W)_{mn} = \frac{\sin 2\pi W(n-m)}{\pi(n-m)}, \quad m, n = 0, 1, \dots, N-1. \quad (2.15)$$

The eigenvectors corresponding to these eigenvalues, normalized to unit energy and arranged according to (2.14), are the DPSS  $\{v_n^{(k)}(N, W)\}$  on the index set  $(0, N-1)$ .

The DPSS are orthogonal in two different index sets. They are orthogonal in  $(0, N-1)$ , i.e.,

$$\sum_{n=0}^{N-1} v_n^{(i)}(N, W) v_n^{(j)}(N, W) = \lambda_i \delta_{ij}, \quad (2.16)$$

and orthonormal in  $(-\infty, \infty)$ ,

$$\sum_{n=-\infty}^{\infty} v_n^{(i)}(N, W) v_n^{(j)}(N, W) = \delta_{ij}. \quad (2.17)$$

The DPSS on the index set  $(0, N-1)$  also satisfies the difference equation

$$\begin{aligned} \frac{1}{2} n (N-n) v_{n-1}^{(k)}(N, W) + \left[ \cos 2\pi W \left( \frac{N-1}{2} - n \right)^2 - \theta_k(N, W) \right] v_n^{(k)}(N, W) \\ + \frac{1}{2} (n+1) [N-1-n] v_{n+1}^{(k)}(N, W) = 0, \end{aligned} \quad (2.18)$$

$$k, n = 0, 1, \dots, N-1$$

where  $\theta_k(N, W)$ 's are the same as in (2.11).

### 2.3 Equations Connecting the DPSWF and the DPSS [11]

Once the DPSS  $\{v_n^{(k)}(N, W)\}$  on the index set  $(0, N-1)$  is known, the DPSWF  $U_k(N, W; f)$  can be computed using the relation

$$U_k(N, W; f) = \epsilon_k \sum_{n=0}^{N-1} v_n^{(k)}(N, W) e^{-j\pi(N-1-2n)f} \quad (2.19)$$

$$k = 0, 1, \dots, N-1$$



where  $\epsilon_k$  is a constant:

$$\epsilon_k = \begin{cases} 1, & k \text{ even} \\ j, & k \text{ odd} \end{cases} \quad (2.20)$$

Equation (2.19), is similar to the discrete Fourier transform equation and thus a similar inverse relation exists:

$$v_n^{(k)}(N, W) = \frac{1}{\epsilon_k} \int_{-1/2}^{1/2} U_k(N, W; f) e^{j\pi(N-1-2n)f} df, \quad (2.21)$$

$$n, k = 0, 1, \dots, N-1.$$

This relates the DPSS to the DPSWF. This inverse relation is useful to find out the DPSS only in the index range  $(0, N-1)$ . For all  $n$ , the following relation holds:

$$v_n^{(k)}(N, W) = \frac{1}{\epsilon_k \lambda_k(N, W)} \int_{-W}^W U_k(N, W; f) e^{j\pi(N-1-2n)f} df, \quad (2.22)$$

$$k = 0, 1, \dots, N-1.$$

Denoting by  $B_w$  and  $I_{N_1}^{N_2}$ , the Band-limiting and Index-limiting operators, such that

$$B_w H(f) = \begin{cases} H(f) & |f| \leq w \\ 0 & |f| > w \end{cases} \quad (2.23)$$

and

$$I_{N_1}^{N_2} \{h_n\} = \begin{cases} h_n & N_1 \leq n \leq N_2 \\ 0 & \text{otherwise} \end{cases} \quad (2.24)$$

the above DPSS-DPSWF interconnecting equations can be represented as the

following transform pairs:

$$\epsilon_k I_0^{N-1} \{v_n^{(k)}(N, W)\} \longleftrightarrow U_k(N, W; f) e^{j\pi(N-1)f}, \quad (2.25)$$

$$\epsilon_k \lambda_k(N, W) \{v_n^{(k)}(N, W)\} \longleftrightarrow B_w U_k(N, W; f) e^{j\pi(N-1)f}. \quad (2.26)$$

These transform equations clearly indicate the fact that band-limiting (and time-limiting) 'stretches' the signal in the time (frequency) domain. Thus in (2.25), index-limiting the DPSS, the transform exists for all  $f$ . In (2.26), band-limiting the DPSWF, the inverse transform exists for all  $n$ .

In the present work, the DPSS are evaluated first, using equation (2.12). The DPSWF are then computed at the necessary frequency points using the interconnecting equation (2.19). The orthogonality property of the DPSWF, equations (2.5) and (2.6), is used to include higher order DPSWF in filter design, and this will be seen in the next Chapter.

## 2.4 Extremal Properties [11]

There are two important extremal properties, that make these discrete prolate functions useful in simultaneous time-limiting and band-limiting. These properties are presented in the following sections. It is on the basis of these properties that these functions find their use in spectral analysis as an optimal filter, and this is discussed in the next section on prolate filter.

### 2.4.1 Most Concentrated Band-limited Sequence

Consider the set of all sequences, whose spectrum vanishes outside  $|f| > W$  and denote them by  $\{h_n\}$ , where the flower brackets stand to identify a sequence (not just the  $n$ th element,  $h_n$ ) and the index  $n$  ranges from  $-\infty$  to  $\infty$ . The energy of the sequence is defined as

$$E \triangleq \sum_{n=-\infty}^{\infty} |h_n|^2 = E(-\infty, \infty) \quad (2.27)$$

The sequence  $\{h_n\}$  has a discrete Fourier transform  $H(f)$ . Since  $\{h_n\}$  is band-limited to  $-W \leq f \leq W$ , the following inverse relation holds:

$$h_n = \int_{-W}^W H(f) e^{-j2\pi n f} df, \quad 0 \leq W \leq 1/2, \quad n = 0, \pm 1, \pm 2, \dots \quad (2.28)$$

Parseval's theorem relates the energy in the two domains as,

$$E = \sum_{n=-\infty}^{\infty} |h_n|^2 = \int_{-W}^W |H(f)|^2 df. \quad (2.29)$$

The energy of the sequence  $\{h_n\}$  on the index set  $(N_0, N_0+N-1)$ , denoted by  $E(N_0, N_0+N-1)$ , is given by

$$E(N_0, N_0+N-1) = \sum_{n=N_0}^{N_0+N-1} |h_n|^2. \quad (2.30)$$

Consider the energy concentration parameter  $\lambda$ , defined as the ratio between  $E(N_0, N_0+N-1)$  and  $E(-\infty, \infty)$ , i.e.,

$$\lambda \triangleq \frac{E(N_0, N_0+N-1)}{E(-\infty, \infty)}. \quad (2.31)$$



Using (2.28) & (2.29),  $\lambda$  can be written as

$$\lambda = \frac{\sum_{n=N_0}^{N_0+N-1} \int_{-W}^W df \int_{-W}^W df' H(f) H^*(f') e^{-j2\pi n(f-f')}}{\int_{-W}^W |H(f)|^2 df}$$

Taking the summation inside the integration (the operations are commutative in this case), we get

$$\lambda = \frac{\int_{-W}^W df \int_{-W}^W df' e^{-j\pi(2N_0+N-1)(f-f')} \frac{\sin N\pi(f-f')}{\sin \pi(f-f')} H(f) H^*(f')}{\int_{-W}^W |H(f)|^2 df} \quad (2.32)$$

This follows from the fact that,

$$\sum_{n=N_0}^{N_0+N-1} e^{-j2\pi n(f-f')} = e^{-j\pi(2N_0+N-1)(f-f')} \frac{\sin N\pi(f-f')}{\sin \pi(f-f')}. \quad (2.33)$$

Let  $\psi(f)$  be defined as,

$$\psi(f) \triangleq e^{j\pi(2N_0+N-1)f} H(f). \quad (2.34)$$

This reduces  $\lambda$  in (2.32) to,

$$\lambda = \frac{\int_{-W}^W df \int_{-W}^W df' \frac{\sin N\pi(f-f')}{\sin \pi(f-f')} \psi(f) \psi^*(f')}{\int_{-W}^W |\psi(f)|^2 df}. \quad (2.35)$$

It can be shown that  $\lambda$  is stationary when  $\psi(f)$  satisfies the DPSWF integral equation (2.1). Thus the maximum value of  $\lambda$  is  $\lambda_0(N, W)$  and this occurs when  $\psi(f) = c U_0(N, W; f)$  for  $|f| \leq W$ , where  $c$  is a scalar constant, independent of  $f$ . From (2.34), we then have

$$H(f) = \begin{cases} c e^{j\pi(2N_0+N-1)f} U_0(N, W; f) & |f| \leq W \\ 0 & W < |f| \leq 1/2 \end{cases} \quad (2.36)$$

Using (2.28), the band-limited sequence  $\{h_n\}$  is

$$h_n = c \int_{-W}^W U_0(N, W; f) e^{j\pi(N-1-2(n-N_0)f)} df \quad (2.37)$$

$n = 0, \pm 1, \pm 2, \dots$

Comparing this to (2.22), the equation connecting the DPSS & DPSWF, we have

$$\{h_n\} = d \{v_{n-N_0}^{(0)}(N, W)\} \quad (2.38)$$

where  $d$  is a scalar constant, independent of  $n$ .

Thus a band-limited sequence, most concentrated in the sense of (2.31), is the DPSS  $\{v_{n-N_0}^{(0)}(N, W)\}$  to within a scalar factor.

#### 2.4.2 Index-limited Sequence with Most Concentrated Spectrum

Let  $\{h_n\}$  be an index-limited sequence, so that

$$h_n = \begin{cases} 0 & n < N_0 \\ h_n & N_0 \leq n \leq N_0 + N - 1 \\ 0 & n > N_0 + N - 1 \end{cases} \quad (2.39)$$

Let the ratio of the energy of its spectrum in the interval  $(-W, W)$ , to the total energy of  $\{h_n\}$ , be represented by  $\mu$ , i.e.,

$$\mu \triangleq \frac{\int_{-W}^W |H(f)|^2 df}{\int_{-1/2}^{1/2} |H(f)|^2 df} \quad (2.40)$$

The spectrum  $H(f)$  is given by the transform equation

$$H(f) = \sum_{n=N_0}^{N_0+N-1} h_n e^{-j n 2\pi f} \quad (2.41)$$

Using this relation, (2.40) can be reduced to

$$\mu = \frac{\int_{-W}^W df \sum_{n=N_0}^{N_0+N-1} \sum_{m=N_0}^{N_0+N-1} h_n h_m^* e^{j 2\pi f (m-n)}}{\sum_{n=N_0}^{N_0+N-1} |h_n|^2}$$

Interchanging orders of summation and integration, we get

$$\mu = \frac{\sum_{n=N_0}^{N_0+N-1} \sum_{m=N_0}^{N_0+N-1} h_n h_m^* \int_{-W}^W e^{j 2\pi f (m-n)} df}{\sum_{n=N_0}^{N_0+N-1} |h_n|^2}$$



After integrating, we get

$$\mu = \frac{\sum_{n=N_0}^{N_0+N-1} \sum_{m=N_0}^{N_0+N-1} h_n h_m^* \frac{\sin 2\pi W(n-m)}{\pi(n-m)}}{\sum_{n=N_0}^{N_0+N-1} |h_n|^2}$$

Finally, translating the indices  $n$  &  $m$  to the  $(0, N-1)$  interval, for convenience,

$$\mu = \frac{\sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \frac{\sin 2\pi W(n-m)}{\pi(n-m)} h_{n+N_0} h_{m+N_0}^*}{\sum_{n=0}^{N-1} |h_{n+N_0}|^2} \quad (2.42)$$

Comparing the above equation to (2.12), the equation that defines the DPSS, it is seen that  $\mu$  will be maximum when

$$\{h_{n+N_0}\} = c \{v_n^{(0)}(N, W)\}, \quad n = 0, 1, \dots, N-1. \quad (2.43)$$

Thus the index-limited sequence, with most concentrated spectrum in  $-W \leq f \leq W$  is,

$$\{h_n\} = \begin{cases} 0 & n < N_0 \\ c v_{n-N_0}^{(0)}(N, W) & N_0 \leq n \leq N_0+N-1. \\ 0 & n > N_0+N-1 \end{cases} \quad (2.44)$$

which is the DPSS  $\{v_{n-N_0}^{(0)}(N, W)\}$  on the index set  $(0, N-1)$ . The concentration of its spectrum in the  $(-W, W)$  interval,  $\mu = \lambda_0(N, W)$ . The spectrum is

related to the DPSWF  $U_0(N, W; f)$  by the equation

$$H(f) = d U_0(N, W; f) e^{j\pi(2N_0+N-1)f} \quad \text{for all } f, \quad (2.45)$$

where  $d$  is a constant independent of  $f$ .

## 2.5 The Discrete Prolate Filter

In the sections up to now, the evolution of the DPSS and the DPSWF, and some of their interesting properties, were seen. The prolate filter was defined, in section 1.3, as having the impulse response sequence  $\{h_n\}$  of (2.44), which is the index-limited DPSS  $\{v_{n-N_0}^{(0)}(N, W)\}$  that has the most concentrated spectrum in  $-W \leq f \leq W$ . The filter is an optimal one, in the sense that it is based on the maximization of energy concentration in the passband of a low-pass configuration.  $N$  refers to the filter length and  $W$  is the normalized bandwidth.  $W$  is the point on the frequency axis, inside of which the energy is maximized, and is normalized since the sampling period  $T$  is assumed to be unity. Hence  $W$  is less than or equal to  $1/2$ , since the Nyquist interval at unit sampling rate is  $(-1/2, 1/2)$ . The normalization equation (2.13) for the DPSS, implies that the filter coefficients  $\{h_n\}$  are normalized to unit energy.

Two major areas, in which the prolate filter would be useful, are edge detection in picture processing and detection of a weak tone, in the presence of a strong nearby one, in harmonic analysis. Both these applications need a narrow main lobe and low side lobe levels — features that can be attained for large  $N$  and small  $W$ , in a prolate filter. Chapter 4 discusses the use of the prolate filter in harmonic analysis.

## 2.6 Computation of the DPSS & DPSWF

First, the DPSS  $\{v_n^{(k)}(N, W)\}$  is computed as the solution to the system of equations (seen earlier),

$$\sum_{m=0}^{N-1} \frac{\sin 2\pi W(n-m)}{\pi(n-m)} v_m^{(k)}(N, W) = \lambda_k(N, W) v_n^{(k)}(N, W), \quad (2.12)$$

$$k = 0, 1, \dots, N-1 \text{ and } n = 0, \pm 1, \pm 2, \dots$$

This equation is the familiar characteristic-value or eigenvalue problem. The eigenvalues  $\lambda_k(N, W)$  can be found as the eigenvalues of the matrix  $\rho(N, W)$ , whose elements are given by

$$\rho(N, W)_{mn} = \frac{\sin 2\pi W(n-m)}{\pi(n-m)}, \quad m, n = 0, 1, \dots, N-1. \quad (2.15)$$

Let the corresponding eigenvectors of this matrix be denoted by  $\{e_j^{(k)}(N, W)\}$ . When  $\{e_j^{(k)}(N, W)\}$  are normalized to unit energy according to

$$\sum_{j=0}^{N-1} [e_j^{(k)}(N, W)]^2 = 1, \quad (2.46)$$

and with a sign convention that makes

$$\sum_{j=0}^{N-1} e_j^{(k)}(N, W) \geq 0 \quad \text{and} \quad \sum_{j=0}^{N-1} (N-1-2j) e_j^{(k)}(N, W) \geq 0, \quad (2.47)$$

for each  $k = 0, 1, 2, \dots, N-1$  (the DPSS normalization conventions), then these eigenvectors are simply the DPSS  $\{v_j^{(k)}(N, W)\}$  on the index set  $(0, N-1)$ .



There are several methods to compute the eigenvalues and eigenvectors of a matrix, such as  $\rho(N, W)$  in (2.15), available in literature. Among these, the widely used methods [27] are 1. the Jacobi method, 2. the Householder's method, 3. the polynomial method and 4. the iteration or power method.

The iterative method is useful when both the eigenvalues and the eigenvectors have to be found. They can be obtained simultaneously, in the process of iteration. When used in conjunction with a method of sweeping, that sweeps out previously determined eigenvalues, all the eigenvalues and eigenvectors of a matrix can be found. This method is useful for matrices of small dimensions, about  $(20 \times 20)$  in size. It is thus limited, in this sense. There is an additional convergence problem, when the largest and the next-to-largest eigenvalues have nearly the same values. This situation occurs in the case of  $\rho(N, W)$  for large  $N$  and  $W$ .

For matrices of larger dimensions, a very efficient method is possible using the Householder's tridiagonal method that reduces a given matrix to a tridiagonal one using orthogonal transformations, by employing an algorithm called the QL algorithm [28], to get the eigenvalues and eigenvectors. This algorithm is limited to symmetric matrices, and thus can be used for  $\rho(N, W)$ . When  $N$  is even, a reduction in the size of  $\rho(N, W)$ , from  $(N \times N)$  to  $(N/2 \times N/2)$  is possible [14], to further reduce the computation times.

The QL algorithm is based on the decomposition of the matrix into the product of an orthogonal matrix and a lower triangular matrix. Routines for the

algorithm, written in FORTRAN, are available in the Eispack Guide [29]. Routines TRED2 and TQL2, are useful to compute the eigenvalues of  $\rho(N, W)$ . The routine TRED2 reduces the real symmetric matrix to a symmetric tridiagonal matrix, and accumulates the orthogonal transformations for use in the routine TQL2, which then determines the eigenvalues and eigenvectors of the symmetric tridiagonal matrix. These routines were used, in the present work, for  $\rho(N, W)$  of order up to 50, and took less than 5 minutes for computation on the VAX 11/780 computer. All eigenvalues took between 2 and 3 iterations, at the maximum, to converge.

Once the DPSS are found, the DPSWF  $U_k(N, W; f)$  can be computed using the connecting equation,

$$U_k(N, W; f) = \epsilon_k \sum_{n=0}^{N-1} v_n^{(k)}(N, W) e^{-j\pi(N-1-2n)f} \quad (2.19)$$

for  $k = 0, 1, 2, \dots, N-1$ . This equation is very straightforward, and the DPSWF can be computed at regular intervals of  $f$ , in the range  $-1/2 \leq f \leq 1/2$ .



## CHAPTER 3

### FOURIER ADDITION OF THE DPSWF

In this chapter, the method to include higher order prolate functions, and the results based on this method, for various  $N$  and  $W$ , are presented. A filter, that uses these higher order functions, is compared to a Chebyshev filter with similar parameters, at the end of the chapter.

#### 3.1 Method for Improvement of Passband Performance

Papoulis and Bertran [13] had suggested the use of a correcting system, in cascade with the prolate filter, to improve the passband performance. In this work, the DPSWF of higher orders are used in conjunction with the order zero  $U_0(N, W; f)$ , to achieve the desired flatness. The flatness is possible, because these functions are used to minimize the mean squared error from the ideal response characteristic. The method uses the orthogonality property of the DPSWF. The idea is simple, and is similar to the Fourier series approximation of arbitrary functions. A discussion of the theory behind Fourier series approximation, is given in the following section.

##### 3.1.1 Least Squares Approximation and Fourier Series [30, 31]

The general linear approximation equation is given by

$$f(t) \sim \sum_{i=1}^n c_i \phi_i(t), \quad (3.1)$$

where  $f(t)$  is the given function, to be approximated in the interval  $[t_1, t_2]$ ;  $\phi_i(t)$ 's are the orthogonal basis functions, in the interval  $[t_1, t_2]$  and



$c_i$ 's are the weighting coefficients. The problem of approximation is one of using the linear weighted combination of  $\phi_i(t)$ ,  $i = 1, 2, \dots, n$ , such that it is close to  $f(t)$  in some sense. The variable  $t$  signifies that a time signal  $f(t)$  is being dealt with, but this is not a restriction. It could be any other physical (or non-physical, for that matter) quantity which is chosen as an independent variable. The criterion used for nearness is the Least-Squares Norm, or  $L_2$ -Norm as it is usually denoted. For this norm, the quantity MSE, given by

$$MSE \triangleq \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \left[ f(t) - \sum_{i=1}^n c_i \phi_i(t) \right]^2 dt, \quad (3.2)$$

called the mean square error, is minimized by proper choice of the  $c_i$ 's. It has been shown [31] that the  $c_i$ 's are given by

$$c_i = \frac{\int_{t_1}^{t_2} f(t) \phi_i(t) dt}{\int_{t_1}^{t_2} [\phi_i(t)]^2 dt}, \quad (3.3)$$

to make MSE minimum. The MSE is then called the minimum mean squared error (MMSE). The  $\phi_i(t)$ 's are square integrable and orthogonal, where orthogonality implies that

$$\int_{t_1}^{t_2} \phi_i(t) \phi_j(t) dt = \begin{cases} 0 & i \neq j \\ k_i & i = j \end{cases} \quad (3.4)$$

for some set of constants  $k_i$ . Orthogonality is not a necessity for (3.1) to be

true. It is enough that the functions  $\{\phi_i(t)\}$  be linearly independent. If they are not orthogonal, however, the expression for  $c_i$ 's, (3.3) is not valid and the evaluation of these weights is more complicated.

Fourier series is used in the approximation of periodic functions. It uses  $\cos kt$  and  $\sin kt$ ,  $k = 1, 2, \dots$ , as basis functions, leading to the approximation equation

$$f(t) \sim \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt). \quad (3.5)$$

These basis functions are periodic with a period of  $2\pi$ . If  $f(t)$  were aperiodic, (3.5) would still hold true in the range  $(-\pi, \pi)$ , and outside this range,  $f(t)$  would be approximated as the repetition of the fundamental form in  $(-\pi, \pi)$ . Thus outside the range  $(-\pi, \pi)$ , the approximation would be in error, but usually the range of interest in approximating an aperiodic function would be finite. Other general intervals, like  $[t_1, t_2]$  can be obtained by proper scaling and shifting of the sinusoids.

### 3.1.2 Approximation Using the DPSWF

The function that is being approximated, is the ideal digital low-pass filter response, viz.,

$$F(f) = \begin{cases} 1 & -W \leq f \leq W \\ 0 & \text{elsewhere in } [-1/2, 1/2]. \end{cases} \quad (3.6)$$

The DPSWF  $U_k(N, W; f)$ ,  $k = 0, 1, 2, \dots, N-1$ , are used as the basis set of functions. The approximation equation is

$$F(f) \sim \sum_{k=0}^{N-1} c_k U_k(N, W; f). \quad (3.7)$$

The  $c_k$ 's can be evaluated on the basis of (3.3), i.e., using:

$$c_k = \frac{\int_{-1/2}^{1/2} F(f) U_k(N, W; f) df}{\int_{-1/2}^{1/2} [U_k(N, W; f)]^2 df}. \quad (3.8)$$

The DPSWF are orthonormal in the  $(-1/2, 1/2)$  interval, and so

$$\int_{-1/2}^{1/2} [U_k(N, W; f)]^2 df = 1, \quad k = 0, 1, \dots, N-1. \quad (3.9)$$

We also have,

$$\int_{-1/2}^{1/2} F(f) U_k(N, W; f) df = \int_{-W}^W U_k(N, W; f) df, \quad (3.10)$$

since  $F(f)$  is unity inside  $(-W, W)$  and vanishes outside this interval. As a result, (3.8) simplifies to

$$c_k = \int_{-W}^W U_k(N, W; f) df. \quad (3.11)$$

The ideal response functions  $F(f)$  is an even function of  $f$  about the  $f = 0$  axis, and so are the DPSWF  $U_k(N, W; f)$  for  $k$  even. Hence the coefficients  $c_k$



for  $k$  odd, are zero in (3.7). This reduces (3.7) to

$$F(f) \sim \sum_{k=0}^{N-1'} c_k U_k(N, W; f) \quad (3.12)$$

where, the prime indicates summation over even indices.

It was seen earlier [Eq. (2.45)] that  $U_0(N, W; f)$  has maximum energy concentration in the interval  $(-W, W)$ . This concentration equals  $\lambda_0(N, W)$ . For the higher order DPSWF  $U_k(N, W; f)$ , the energy concentration in the interval  $(-W, W)$  is given by  $\lambda_k(N, W)$ . Since the  $\lambda_k(N, W)$ 's are ordered such that it decreases with increasing index  $k$ , [Eq. (2.3)], the passband [the  $(-W, W)$  interval] energy concentration for higher order DPSWF  $U_k(N, W; f)$  decreases with  $k$ . As a result, the total energy concentration in the passband, for the linear weighted sum (3.12), decreases as higher order DPSWF are included. From (3.12), it can be seen that the total energy concentration  $\lambda'(N, W)$  is given by

$$\begin{aligned} \lambda'(N, W) &= \frac{c_0 \lambda_0(N, W) + c_2 \lambda_2(N, W) + \dots}{c_0 + c_2 + \dots} \\ &= \frac{\sum_{k=0}^{N-1'} c_k \lambda_k(N, W)}{\sum_{k=0}^{N-1'} c_k} \end{aligned} \quad (3.13)$$

Thus  $\lambda'(N, W)$  is less than  $\lambda_0(N, W)$ . The energy lost in the passband is transferred to the stopband and the transition region between the two bands.

This leads to widened main lobe and/or increased side lobe levels in the magnitude spectrum of the filter designed according to (3.12).

In (3.6),  $F(f)$  is a periodic function of  $f$ , with a period of 1. The DPSWF for  $k$  even, are even functions of  $f$  about  $f = 0$ . They are also periodic in  $f$ . Their period is 1, when  $N$  is odd and 2 when  $N$  is even.  $F(f)$  can also be chosen to span the same period. The approximation (3.12), is similar to the classical Fourier series representation of periodic functions. For this reason, the filter designed using (3.12), shall be called a Fourier prolate filter, from now on.

### 3.2 Results and Discussion

In the sections that follow, some of the results based on (3.12) are presented, for selected values of  $N$  and  $W$ , and one of the cases is compared with a Chebyshev design of same filter length  $N$  and similar passbands and stopbands.

#### 3.2.1 Filters with $N = 4$ and $N = 5$ , for $W = 0.20$

Fig. 1. compares the prolate and Fourier prolate filters for the set of values,  $N = 4$  and  $W = 0.20$ . There are four orders of DPSWF  $U_k(N, W; f)$ , viz.,  $k = 0, 1, 2$  and  $3$ . Of these,  $U_0(N, W; f)$  and  $U_2(N, W; f)$  are even functions of frequency, and hence these are the only ones of interest. The prolate filter is simply characterized by  $U_0(N, W; f)$ . The Fourier prolate filter is formed using a weighted sum of  $U_0(N, W; f)$  and  $U_2(N, W; f)$  according to (3.12).

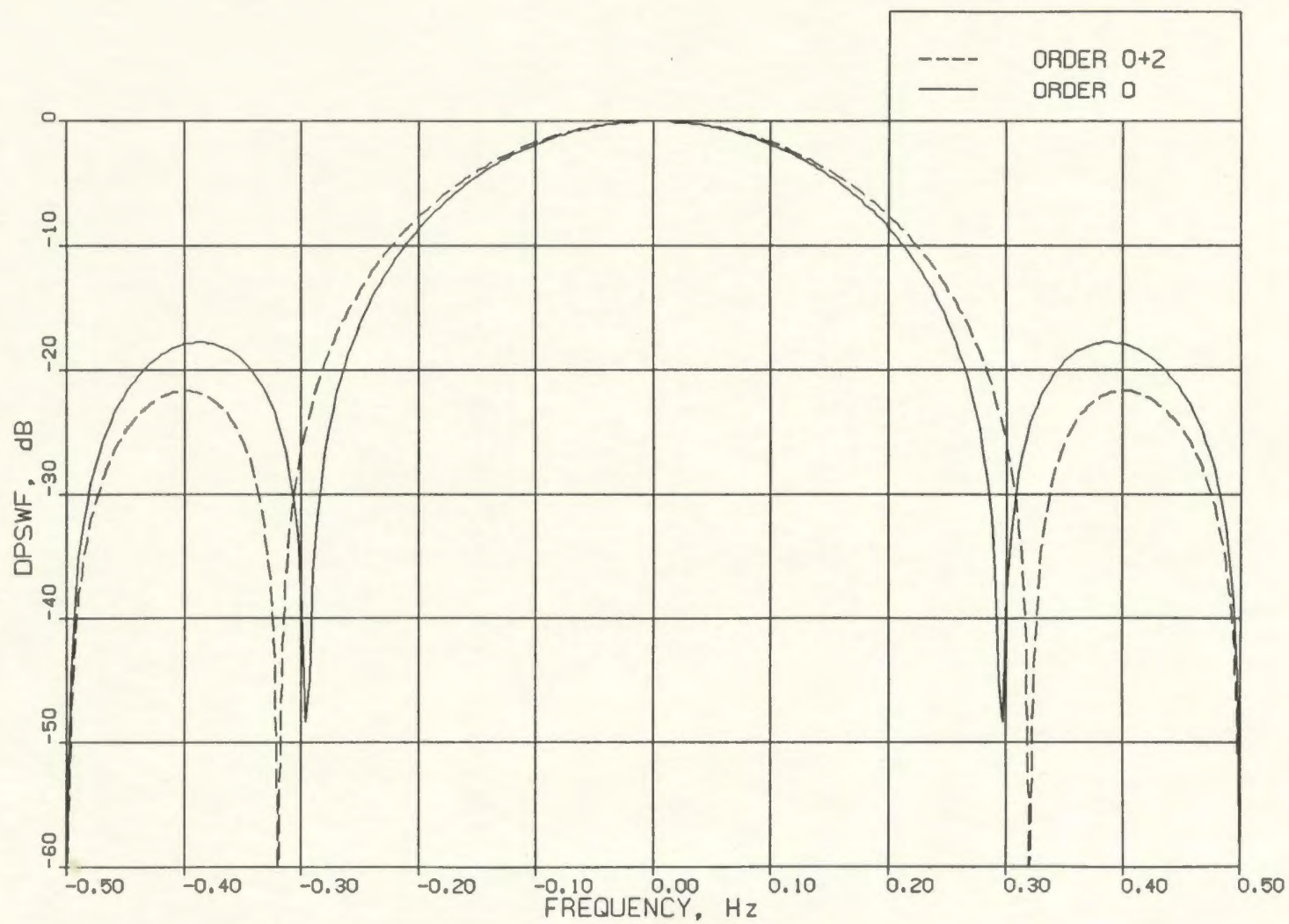


FIG. 1 -  $N = 4$ ,  $W = 0.20$   
FOURIER ADDITION OF THE DPSWF



It can be observed that  $U_0(N, W; f)$  has a main lobe and a side lobe in the frequency interval  $(0, 1/2)$ . The addition of  $U_2(N, W; f)$  has widened the main lobe and has brought the side lobe level down by about 5 dB. There is a slight improvement in the passband performance. This can be measured, to some extent, by looking at the edge-of-the-passband gain. At this edge, i.e. at  $f = W = 0.20$ , there is an increase in gain of about 1 dB in the response curve. This is only a very slight improvement, but that is expected for small  $N$ .

The performances of the prolate and Fourier prolate filters with  $N = 5$  and  $W = 0.20$ , are compared in Fig. 2. Now  $k$  can be 0, 1, 2, 3 and 4, or there are three basis functions  $U_0(N, W; f)$ ,  $U_2(N, W; f)$  and  $U_4(N, W; f)$  that are even functions of frequency and thus useful for the Fourier prolate filter design. The prolate filter based on  $U_0(N, W; f)$  alone, has a main lobe and two side lobes in the interval  $(0, 1/2)$ . To this is added  $U_2(N, W; f)$ , the next even function. Again this results in a widened main lobe, 5 dB gain improvement at the edge of the passband, and just one side lobe which is 10 dB below the prolate filter's first side lobe.  $U_4(N, W; f)$  is then added to the other two functions (again in accordance with (3.12)), and it should be noted that there is virtually no difference in performance. This is because the weighting coefficient  $c_4$  is small, since the energy in  $(-W, W)$  is small for  $U_4(N, W; f)$ .

In both these cases, the transfer of passband energy is to the transition region, and this has lead to widened main lobe. The side lobe levels are lower, but when compared to a prolate filter having that widened main lobe width, these levels would be higher.

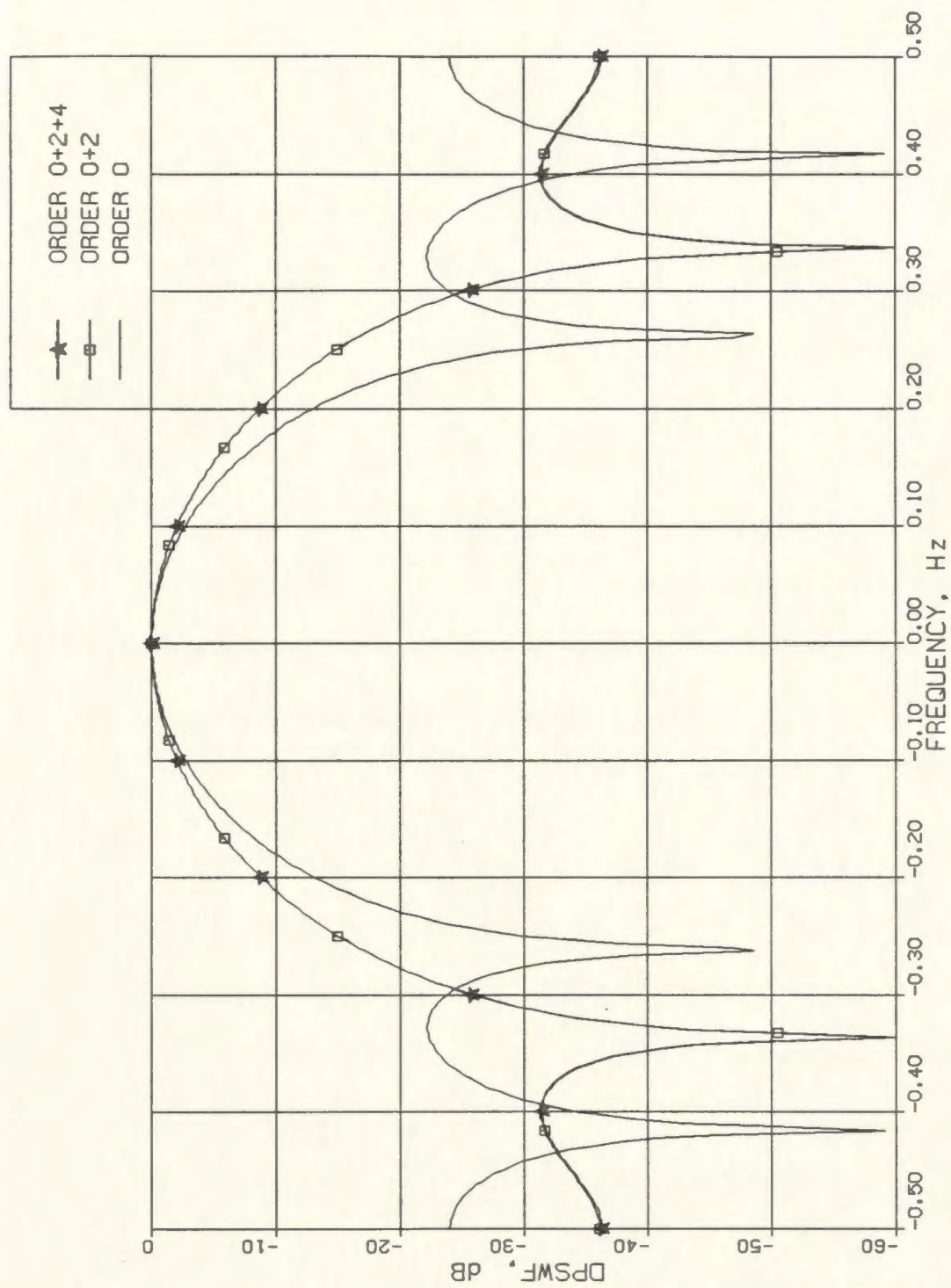


FIG. 2 -  $N = 5$ ,  $W = 0.20$   
FOURIER ADDITION OF THE DPSWF



### 3.2.2 Filters with $N = 6$ and $N = 7$ , for $W = 0.20$

For  $N = 6$  and  $W = 0.20$ , there are still only 3 orders that are useful in the design of the Fourier prolate filter, viz.,  $U_0(N, W; f)$ ,  $U_2(N, W; f)$  and  $U_4(N, W; f)$ . Again the main lobe widens with the addition of  $U_2(N, W; f)$  (see Fig. 3). The edge-of-the-passband response is up by 10 dB, but the side lobe level has gone up by 5 dB. The energy lost in the passband, is transferred to both the transition region and the stopband. As a result, there is a widened main lobe and raised side lobe levels.

The gain at the passband edge is a good measure of flatness, and any improvement at that frequency can lead to an overall improved passband performance. This effect is more pronounced in the next example, where  $N = 7$  (Fig. 4). Clearly, a filter with a flatter passband performance can be seen. In this case, the gain at the edge of the passband is 15 dB better than  $U_0(N, W; f)$ , when  $U_2(N, W; f)$  is added to it. The side lobe level also rises by 15 dB. This tendency of increased side lobe levels, clearly shows that the situation of improvement in the passband performance to that of the reduction of side lobe levels, is a compromising one. Of course, both can be improved simultaneously, as one increases the length  $N$ , of the filter. But for a fixed  $N$ , improvement in one can be achieved at the cost of the other. The passband energy concentration drops from about 99% to 82%, which may not be a serious loss for the kind of applications that warrant a flat passband. Moreover, the side lobes are still 20 dB below the d.c. gain, and this level is comparable to other filters (of same length) that



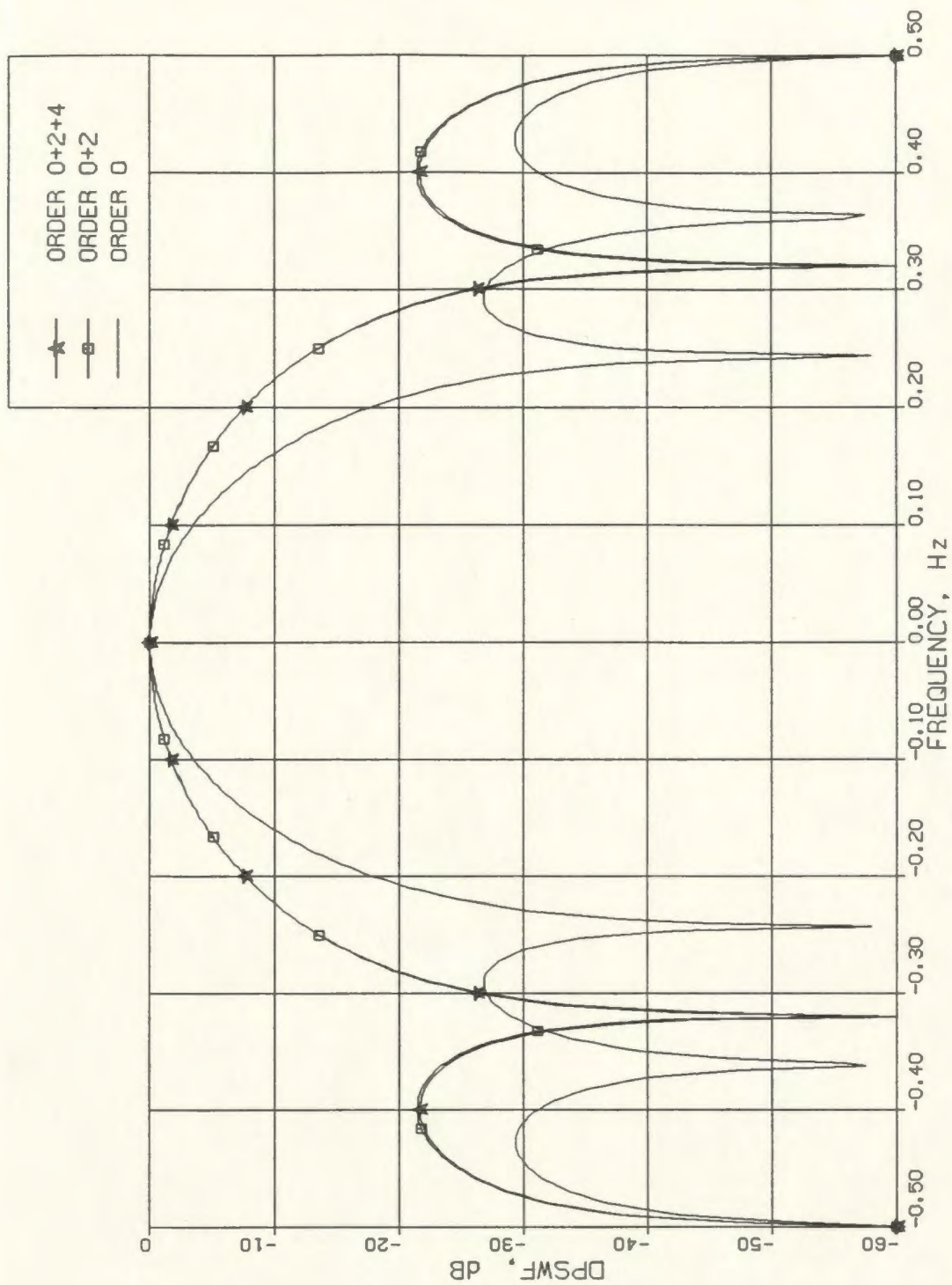


FIG. 3 -  $N = 6$ ,  $W = 0.20$   
FOURIER ADDITION OF THE DPSWF

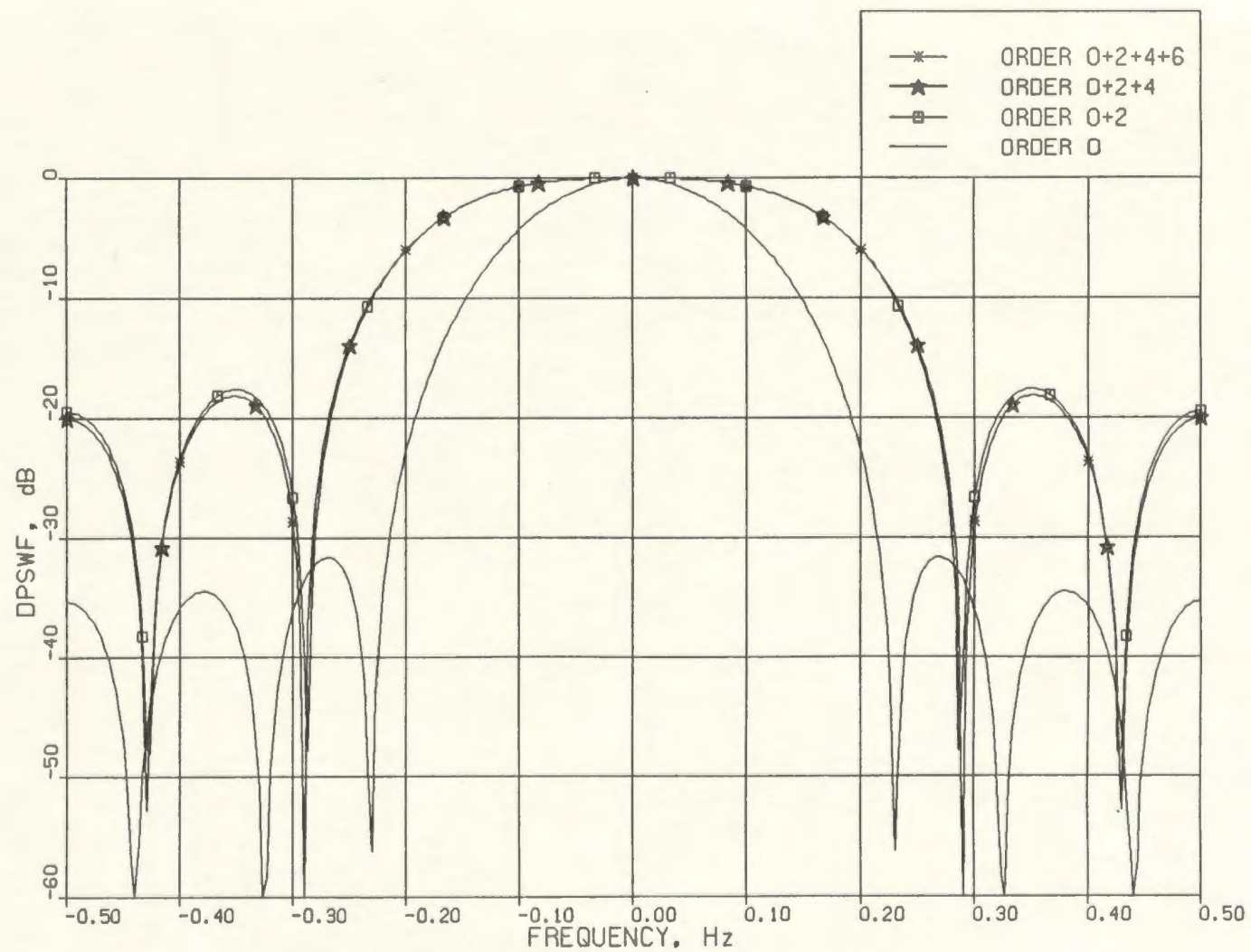


FIG. 4 -  $N = 7$ ,  $W = 0.20$   
FOURIER ADDITION OF THE DPSWF

employ a flatness-in-the-passband criterion. The addition of  $U_4(N, W; f)$  does not make much improvement in the passband performance of the Fourier prolate filter.

### 3.2.3 Filter with $N = 10$ and $W = 0.20$

Next, consider a filter of order  $N = 10$ , and  $W = 0.20$ . The effects of including one by one, the orders  $U_2(N, W; f)$ ,  $U_4(N, W; f)$ ,  $U_6(N, W; f)$  and  $U_8(N, W; f)$  can be seen in Fig. 5, 6 and 7, on the following pages. The prolate filter  $U_0(N, W; f)$  has 4 side lobes in the interval  $(0, 1/2)$ . The highest side lobe is at a level of about  $-45$  dB, and at the passband edge,  $f = 0.20$ , the gain is about  $-35$  dB. With the addition of  $U_2(N, W; f)$ , the first side lobe rises up to  $-20$  dB (Fig. 5), but at the edge of the passband there is a gain of  $-10$  dB, which is 25 dB better than the prolate filter. This is a very good improvement. The loss of energy is about 15% within the passband, and there are now 3 side lobes in  $(0, 1/2)$ . A ripple or an overshoot in the amplitude response, of about 3 dB height, can be seen within the passband. This shows that the gain at the passband edge alone, is not a criterion enough to measure flatness in the passband. The ripple amplitude, as well as the average slope in the passband response should be small to achieve flatness. With the inclusion of  $U_4(N, W; f)$ , the side lobe levels get about 5 dB better, and the highest side lobe goes to  $-25$  dB from the  $-20$  dB before; the edge-of-the-passband gain also improves by about 5 dB and rises to  $-5$  dB now; the ripple height reduces to a 2 dB level. There is thus an improvement in every sense, with the addition of  $U_4(N, W; f)$  in the design. Inclusion of the higher orders, viz.,  $U_6(N, W; f)$  and  $U_8(N, W; f)$  does



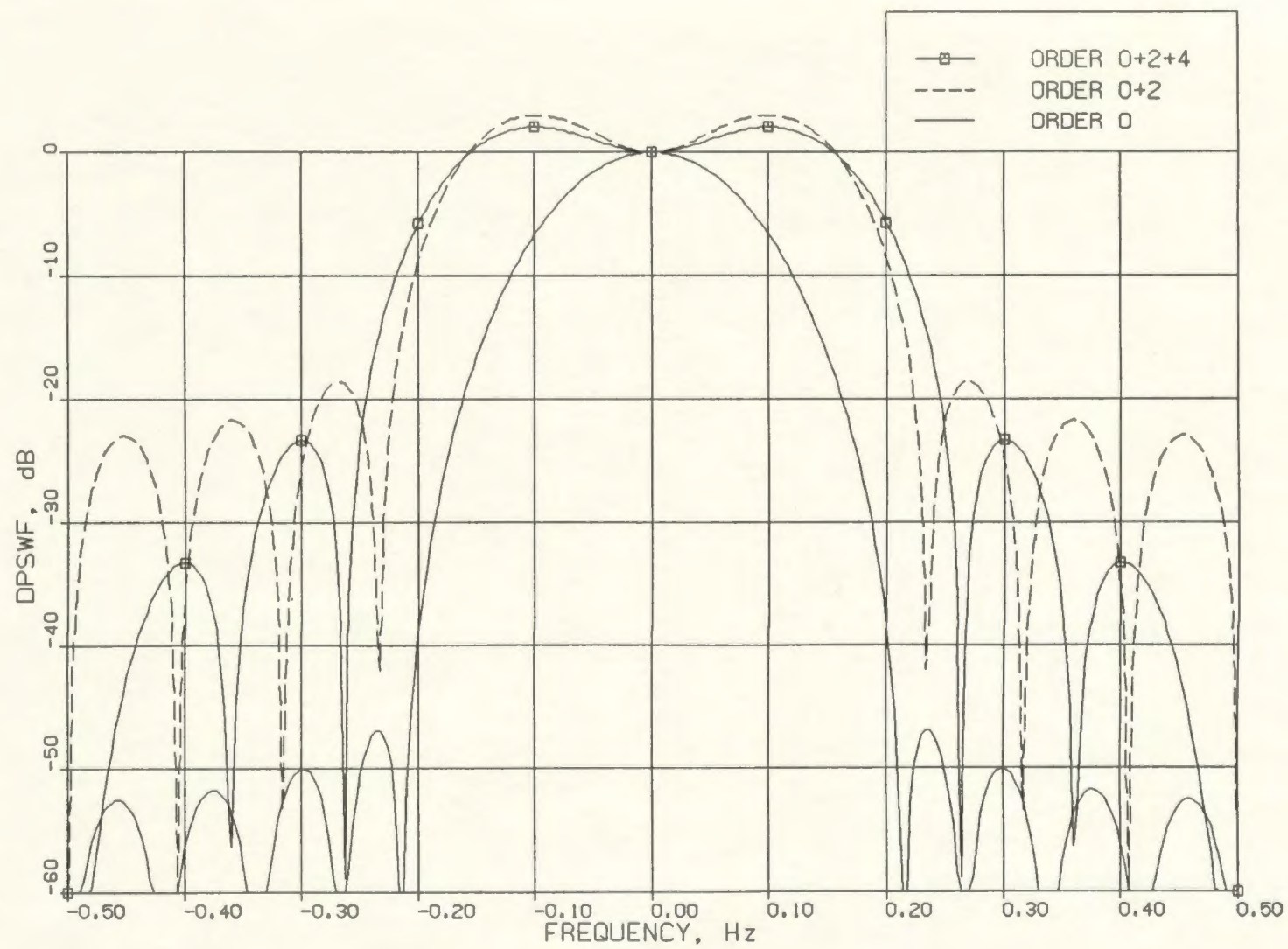


FIG. 5 -  $N = 10$ ,  $W = 0.20$   
FOURIER ADDITION OF THE DPSWF

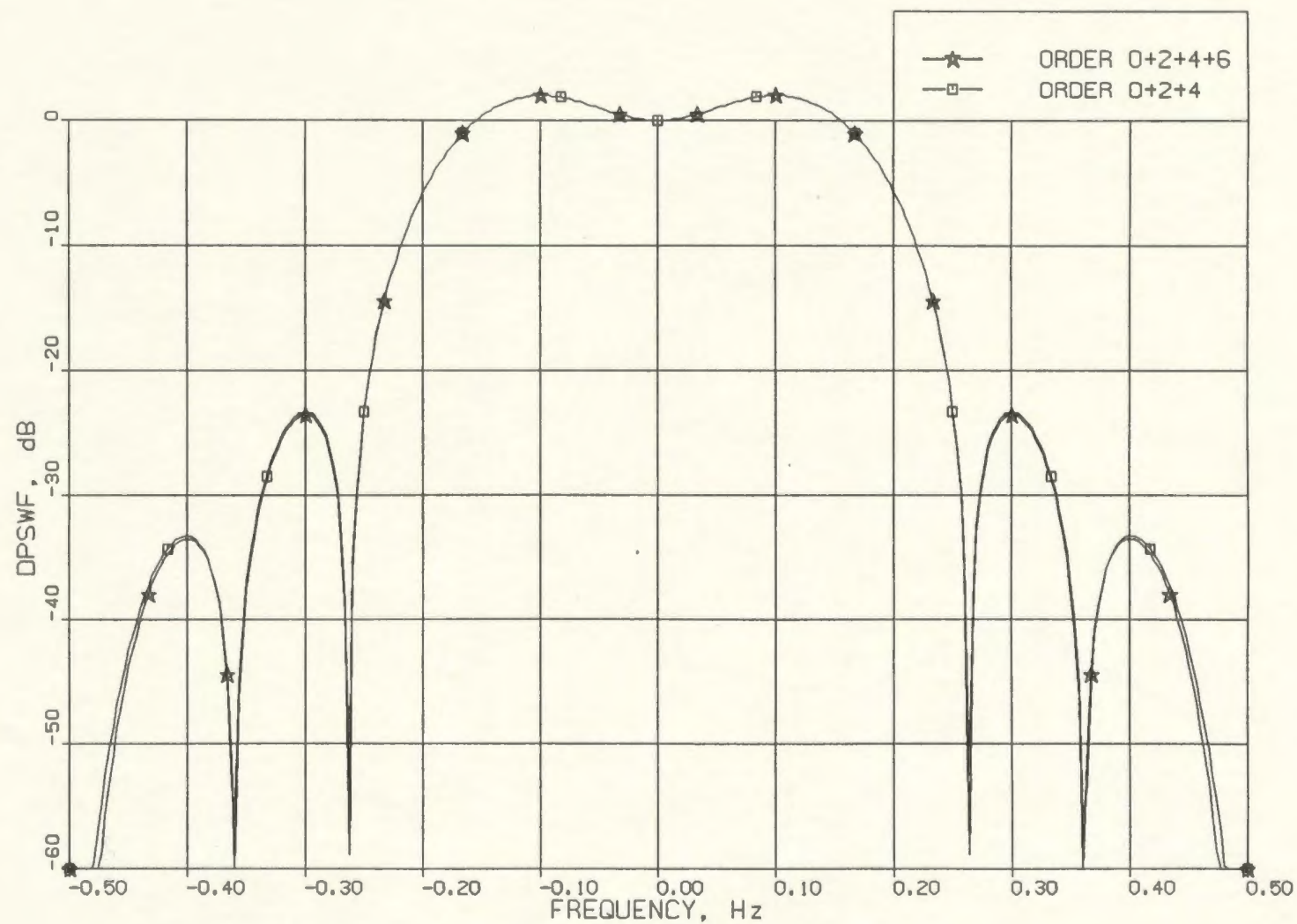


FIG. 6 -  $N = 10$ ,  $W = 0.20$   
FOURIER ADDITION OF THE DPSWF

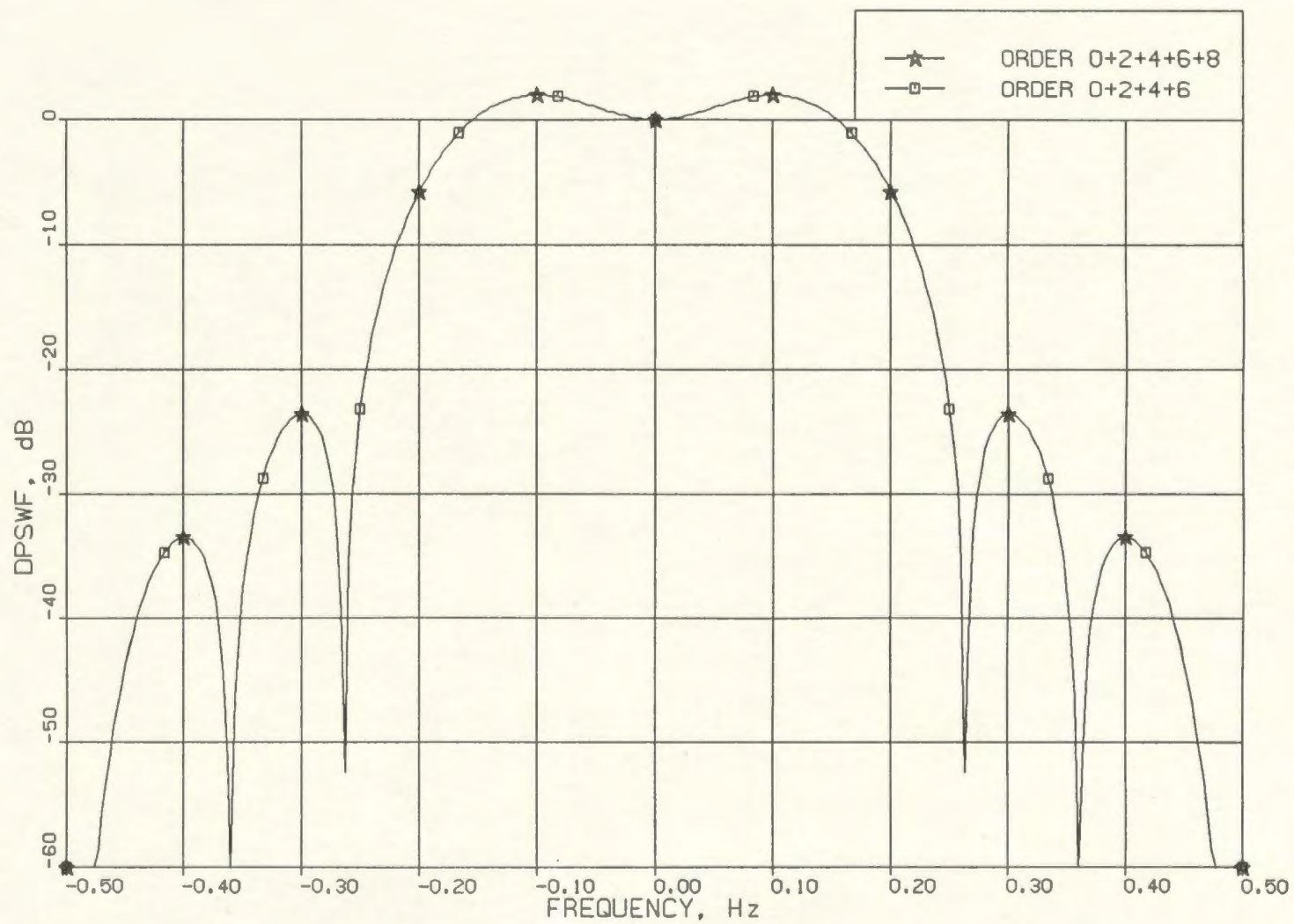


FIG. 7 -  $N = 10$ ,  $W = 0.20$   
FOURIER ADDITION OF THE DPSWF



not make much difference to the design, since their weighting coefficients are small as a result of small energy concentrations in their passband.

### 3.2.4 Filter with $N = 10$ and $W = 0.15$

The widening of the main lobe implies that the resulting Fourier prolate filter has a higher bandwidth (the region within which it has the same energy concentration as that of the prolate filter in  $(-W, W)$ ) compared to that of the prolate filter. It is higher than the normalized bandwidth  $W$ . As a result, for a filter of required bandwidth one can start from the prolate functions of smaller  $W$ . To get a filter of approximately  $W = 0.20$  bandwidth, an initial value of  $W = 0.10$  was tried. After gradual increases, it was found that  $W = 0.15$  gave the required bandwidth (Fig. 8).

The first side lobe of the prolate filter, ( $N = 10, W = 0.15$ ), is at a level of  $-35$  dB. This is 10 dB higher than that of the prolate filter with  $W = 0.20, N = 10$ . This is expected, since the eigenvalues  $\lambda_0(N, W)$  reduce with  $W$  [11], implying that the passband energy concentration is lower for lower  $W$ . This results in an increased energy in the stopband, and hence increased side lobe levels result. Here again, the Fourier prolate design requires the first three even-ordered prolate functions,  $U_0(N, W; f)$ ,  $U_2(N, W; f)$  and  $U_4(N, W; f)$ . The passband ripple is now reduced to nearly 0.5 dB in amplitude (see Fig. 11). The highest side lobe level is  $-15$  dB. This is 10 dB inferior to the previous case, with  $W = 0.20$ , but this is expected since the basis functions themselves have higher side lobe levels in comparison to the previous case. The Fourier prolate design has a bandwidth of about 0.20; it extends roughly between 0.00 and 0.20.

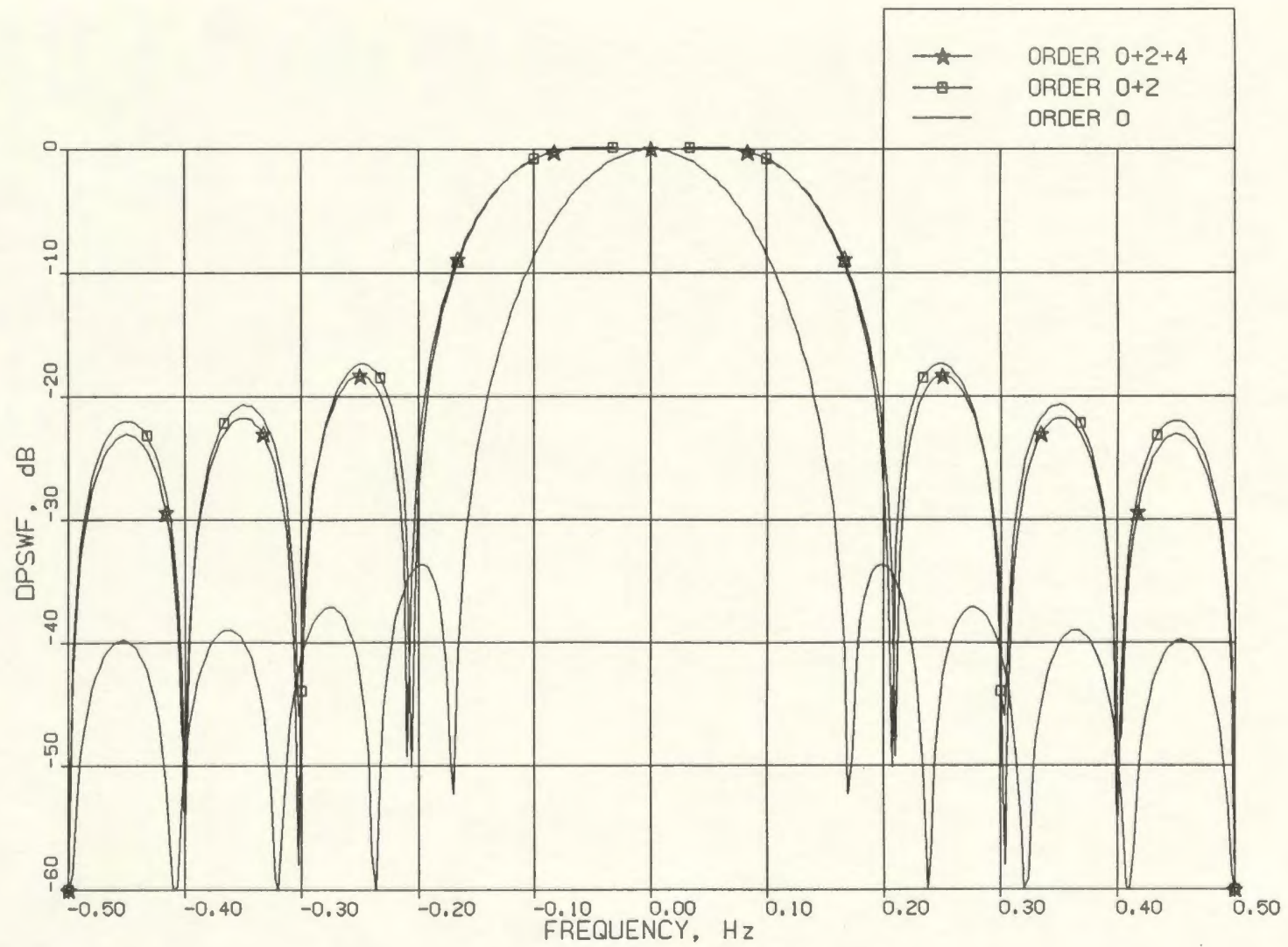


FIG. 8 -  $N = 10$ ,  $W = 0.15$   
FOURIER ADDITION OF THE DPSWF



Figure 9 compares  $U_0(N, W; f)$  with  $N = 10$ ,  $W = 0.20$ , to the Fourier prolate filter using  $U_0(N, W; f)$ ,  $U_2(N, W; f)$  and  $U_4(N, W; f)$  with  $N = 10$  and  $W = 0.15$ . Both filters have a passband extending between 0.00 and approximately 0.20 (on the positive frequency side). The Fourier prolate filter has a much flatter passband, compared to the prolate filter. However, this is at the cost of a side lobe structure, which now has a -20 dB high side lobe, 30 dB inferior to that of the prolate filter.

### 3.2.5 Comparison to the Chebyshev Filter

Till now, the effect of including the higher order DPSWF on the filter response characteristic was seen in comparison to the prolate filter, that is based on the DPSWF  $U_0(N, W; f)$  alone. The Fourier prolate design is compared to a filter, of similar nature and based on the minimax criterion — the standard and highly useful Chebyshev filter.

A Chebyshev filter of order  $N = 10$ , with a passband between  $f = 0$  and 0.20 (again, for the positive side) and a stopband from  $f = 0.25$  to 0.50, is compared to the Fourier prolate filter designed with  $N = 10$  and  $W = 0.15$ . The Chebyshev design was obtained using the FORTRAN routine in [32, pp. 187-204]. The comparison is shown in Fig. 10, and it can be observed that the Fourier prolate design is quite close in performance, to the Chebyshev filter. Both filters have nearly the same side lobe levels and side lobe structures; the edge-of-the-passband gain is also nearly the same. An expanded look at the passband



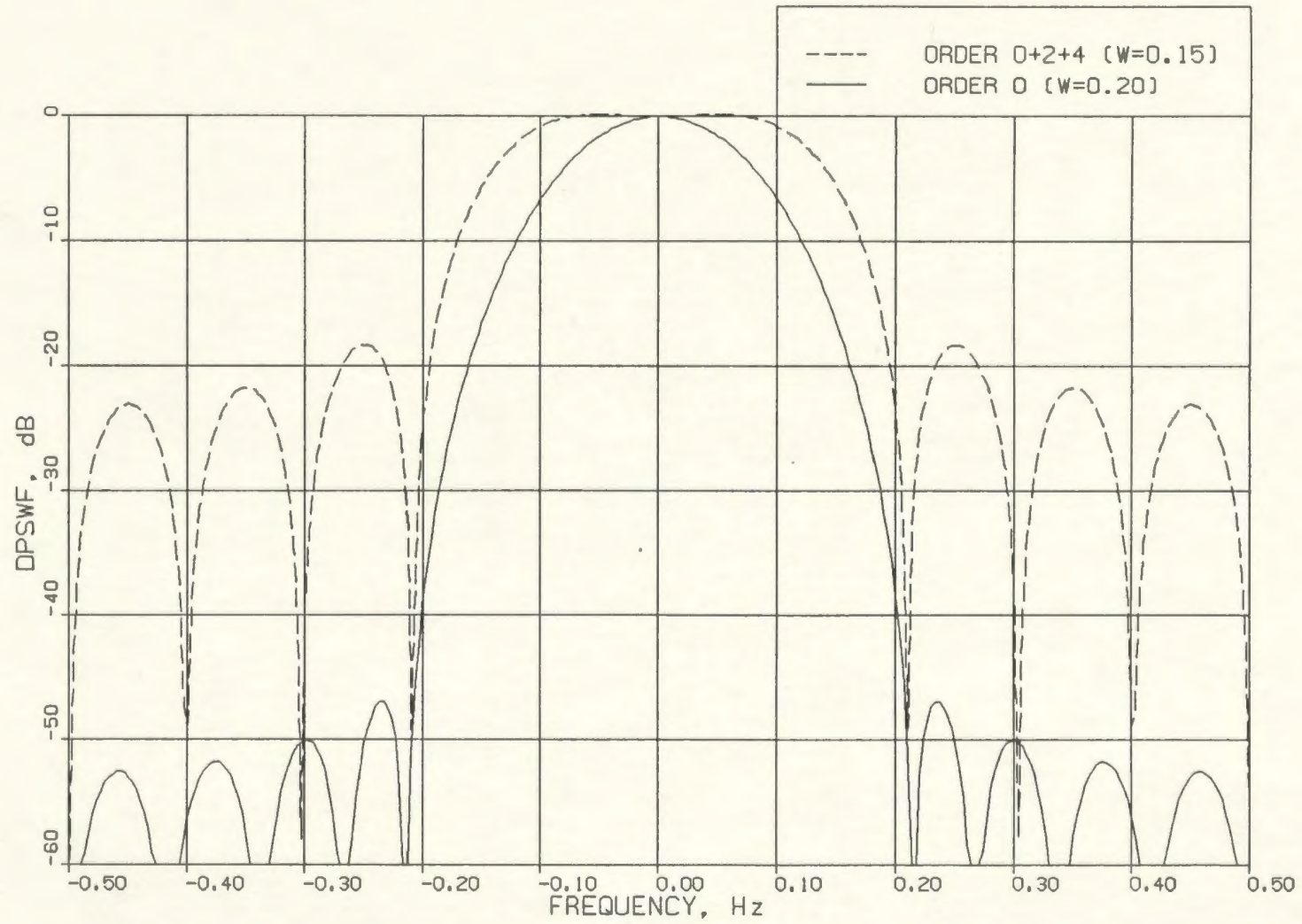


FIG. 9 -  $N = 10$   
FOURIER ADDITION OF THE DPSWF

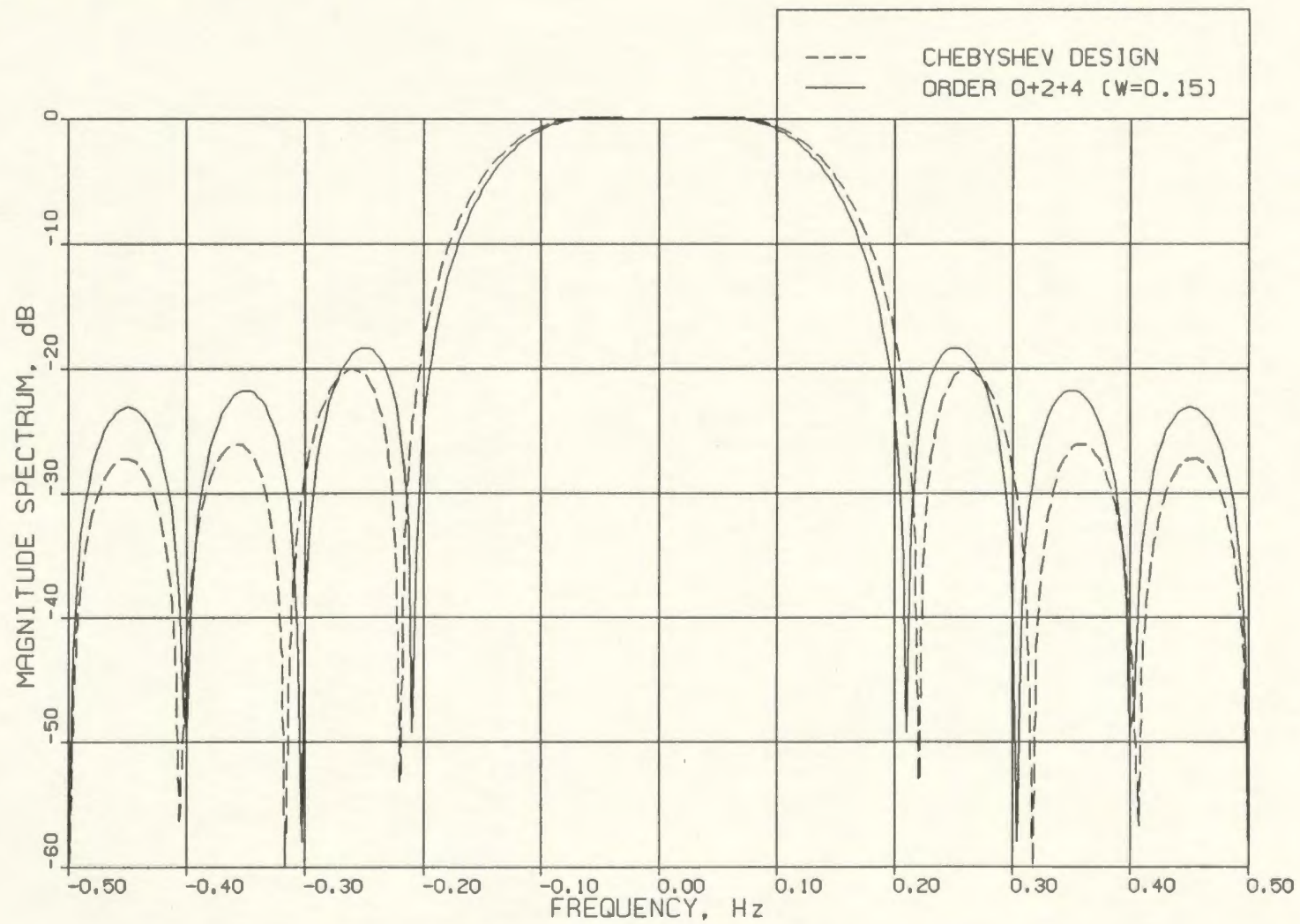


FIG. 10 -  $N = 10$   
COMPARISON OF FOURIER PROLATE & CHEBYSHEV FILTERS

performance shows that both filters have very similar ripple levels (Fig. 11). Thus the Fourier prolate design, in this case, is as good as the Chebyshev design.

### 3.2.6 Summary of Comparisons

The characteristics of three filters, viz., the Fourier prolate filter using orders 0, 2 and 4 of the DPSWF  $U_k(N, W; f)$  with  $N = 10$  and  $W = 0.15$  (Sec. 3.2.4), the prolate filter using DPSWF  $U_0(N, W; f)$  with  $N = 10$  and  $W = 0.20$  (Sec. 3.2.3), and the Chebyshev filter of Sec. 3.2.5, are compared in the following table. The three filters have a main lobe extending from  $f = 0$  to  $f = 0.20$ . It can be seen that the prolate filter has a better side lobe level, but the Fourier prolate and Chebyshev filters have a better gain at the passband edge. The Fourier prolate and Chebyshev filters have nearly equal response levels.

FILTERS	Highest Side Lobe Level (dB)	Gain at the Passband Edge (dB)	Ripple Amplitude (dB)
Prolate	- 46	- 38	0.0
Fourier Prolate	- 18	- 22	0.5
Chebyshev	- 20	- 18	1.0



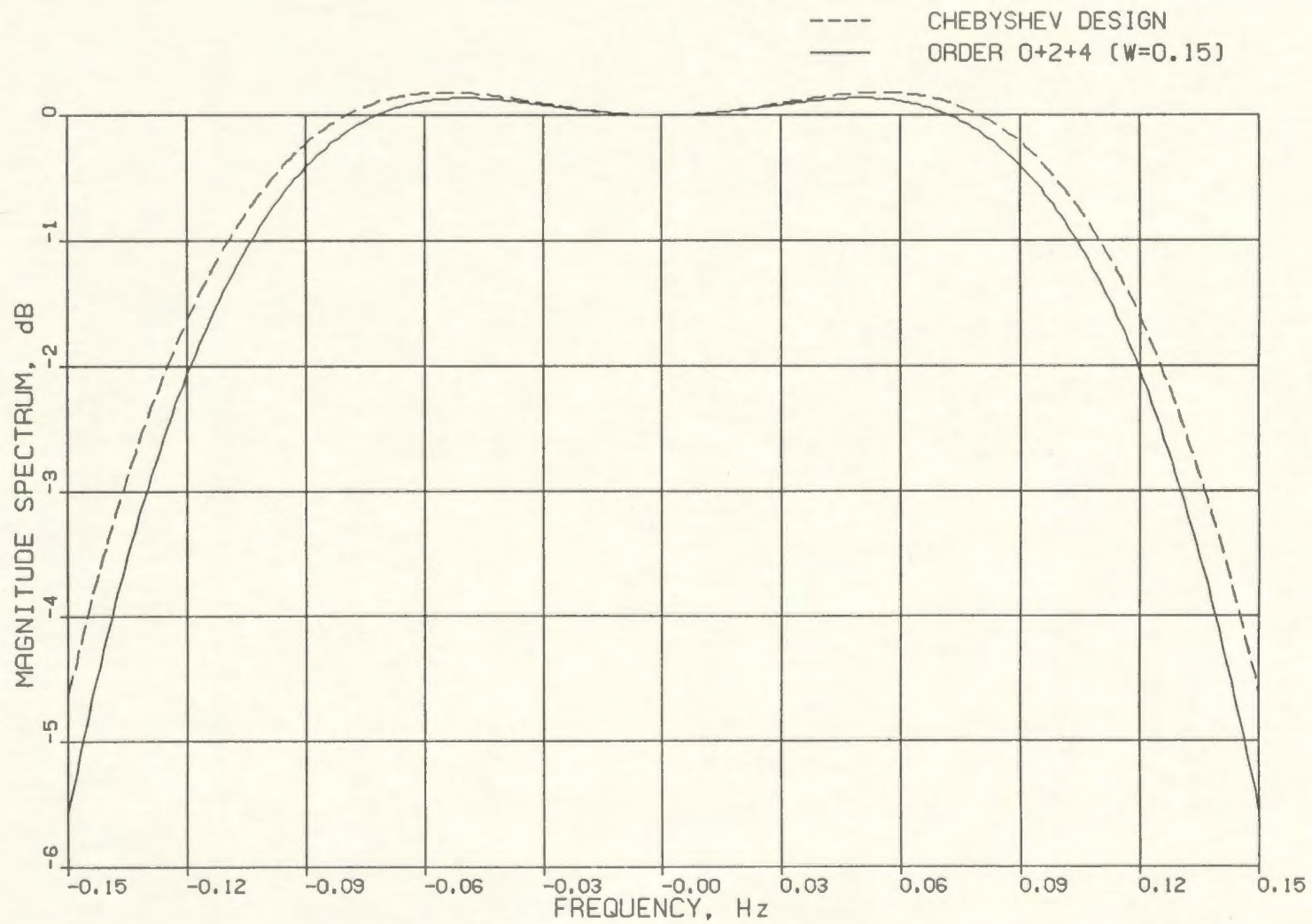


FIG. 11 -  $N = 10$   
PASSBAND PERFORMANCE OF THE FILTERS IN DETAIL

## CHAPTER 4

### PROLATE FUNCTIONS IN THE LIGHT OF HARMONIC ANALYSIS

In this chapter, the role of the DPSS as a window, used in harmonic analysis, is discussed. To this effect, the first discussion is a brief account of the windowing concepts in filter design. This is followed by a look at the use of windows in harmonic analysis. Certain quantities that are useful in comparing various windows, are then defined. The window based on the DPSS, is compared to a few other windows on the basis of the data provided by Harris [25]. Next, a problem of weak-signal detection, in the presence of interference, similar to the example given in [25], is presented. The performance of the prolate window, in this detection problem, is compared to that of the Blackman-Harris window.

#### 4.1 Use of Windows in Filter Design [32]

Linear phase FIR filters can be designed using one of the following three well-known methods [32]: the frequency sampling method, the window method, and optimal design method, such as the one based on Chebyshev criterion. The one optimal design that was seen previously, was the one based on the discrete prolate functions, where the criterion was one of maximizing the passband energy concentration. Each of these three techniques is useful in designing filters for use in a particular application and hence each technique is important. The design technique, that is going to be looked at, is the window technique, as this introduces the concept of windowing, which is very essential in harmonic analysis.

The frequency response  $H(e^{j\theta})$ , of any digital filter is a periodic function of the frequency and can thus be expanded in a Fourier series expression of the form

$$H(e^{j\theta}) = \sum_{n=-\infty}^{\infty} h_n e^{-jn\theta} . \quad (4.1)$$

The Fourier series coefficients  $h_n$  are given by

$$h_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\theta}) e^{jn\theta} d\theta . \quad (4.2)$$

The sequence  $\{h_n\}$ , is the impulse response sequence of the filter. The filter in (4.1) is unrealizable since the impulse response sequence is infinite in duration (as the summation in (4.1) extends to  $\pm\infty$ ). Moreover, an infinite amount of delay is needed to realize the impulse response that starts at  $-\infty$ . Thus the filter coefficients  $h_n$  in the Fourier series representation (4.1), of  $H(e^{j\theta})$  is not realizable.

An FIR design, which would be an approximation of  $H(e^{j\theta})$ , can be obtained by truncating the infinite summation in (4.1) to  $n = \pm N$ . Such a direct truncation of the series leads to Gibbs oscillations [32] in the desired response. The sharp cutoff points in the desired response represent a discontinuity in  $H(e^{j\theta})$ . Its truncated Fourier series gives rise to fixed amplitude overshoot and ripple at the desired sharp cutoff points. Thus in approximating an ideal low-pass filter by a truncation of (4.1), a ripple of maximum amplitude which is



9% of the step size of the ideal response, results in the design. Increasing  $N$  only reduces the frequency range in which the ripple occurs, but not the ripple amplitude. Hence direct truncation of (4.1) is not very useful to design a reasonable FIR filter.

A finite duration weighting sequence  $\{w_n\}$ , called a window, is used to modify the Fourier coefficients  $h_n$  in (4.1), to improve the convergence of the Fourier series. This is called the windowing technique and leads to very reasonable filter designs, and easier realizability in terms of filter length is possible. An FIR approximation to  $H(e^{j\theta})$  is obtained by multiplying the two sequences  $\{h_n\}$  and  $\{w_n\}$ . A new sequence  $\{\hat{h}_n : \hat{h}_n = h_n \cdot w_n\}$  is thus obtained, and this sequence  $\{\hat{h}_n\}$  is finite in duration since  $w_n$  is zero outside a certain interval, say  $-N \leq n \leq N$ .

If  $W(e^{j\theta})$  and  $\hat{H}(e^{j\theta})$  are the transform pairs of  $\{w_n\}$  and  $\{\hat{h}_n\}$  respectively, then  $\hat{H}(e^{j\theta}) = H(e^{j\theta}) * W(e^{j\theta})$ , where  $*$  denotes circular convolution. This follows from the fact that multiplication in one domain corresponds to convolution in the other (transform) domain. For physical realizability and causality,  $\{\hat{h}_n\}$  is shifted to lie in  $(0, 2N)$ . This only affects the phase spectrum by a linear shift, and leaves the magnitude spectrum unperturbed.

Multiplying  $\{h_n\}$  by such a weighting sequence  $\{w_n\}$ , finite in duration, has the effect of smoothing the discontinuities in  $H(e^{j\theta})$  into transition bands at

the region of discontinuities. Since this frequency response results from the circular convolution of both  $H(e^{j\theta})$  and  $W(e^{j\theta})$ , the transition bandwidth depends on the main lobe width of  $W(e^{j\theta})$ . Also the side lobes of  $W(e^{j\theta})$  causes errors in the form of ripples in the response  $\hat{H}(e^{j\theta})$ . As a result, it can be inferred that a good window should have

- (i) small main lobe width, to reduce transition bandwidth, and
- (ii) small side lobes with a fast falloff, so that the side lobes decrease in size as  $\theta \rightarrow \pi$ . This reduces approximation errors in the desired response.

Many windows are available in literature [25]. Basic among them are the rectangle and triangle windows. The popular ones include the Hamming, Hanning and Kaiser windows. The following section describes how the window sequences are useful in harmonic analysis.

## 4.2 Use of Windows in Harmonic Analysis

Harmonic analysis, involves the processing of time waveform or samples to check the spectral contents, and to plot them using relative magnitude levels against the frequency. With respect to this application, windows are applied to the waveform or samples in order to smooth its spectrum.

### 4.2.1 The Discrete Fourier Transform

A time waveform can be of infinite duration in theory, but in practice it is always a finite length of data that is used in spectral analysis. Moreover, for



practicality the data that is processed is an uniformly sampled one having a fixed number,  $N$ , of samples. The spectral analysis is simply the Fourier transform analysis of the signal. For a general signal, this transform exists for all  $\omega$ , where  $\omega$  is the angular frequency variable. It is not practical to calculate the transform for the continuum of all  $\omega$ . It is customary to represent the transform by a set of frequency samples, scanning a finite frequency range.

The equations related to spectral analysis [30], are now presented. A time waveform  $f(t)$  has a Fourier transform  $F(j\omega)$  given by

$$F(j\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt . \quad (4.3)$$

To compute the above integral,  $f(t)$  is represented by a set of samples  $f_k = f(kT)$ . This reduces the above integration to a summation, given by

$$F(e^{j\theta}) = \sum_{k=-\infty}^{\infty} f_k e^{-jk\theta} . \quad (4.4)$$

This is similar to (4.1), and  $F(e^{j\theta})$  denotes the spectrum of the sequence  $\{f_k\}$ , where  $\theta$  is the normalized frequency variable ( $\theta = \omega T$ ). The spectrum  $F(e^{j\theta})$  is periodic with a period of  $2\pi$  (from the sampling theorem). The process of sampling in the time domain leads to periodicity of the spectrum in the frequency domain, the period being  $1/T$  Hz., for a sampling interval of  $T$  seconds.

The Fourier series representation of a periodic function  $f(t)$  gives rise to line spectrum or discrete frequency spectrum (the Fourier coefficients give the



amplitude of each frequency component). This is similar to sampling in the time domain, which leads to a periodic continuous spectrum in the frequency domain. Fourier series can thus be looked upon as sampling in the frequency domain, which leads to a periodic time waveform.  $F(j\omega)$  in (4.3) is represented by its samples  $F_m = F(jm\Omega)$ , and  $f(t)$  by its samples  $f_k = f(kT)$ . Relating these samples by a transformation such as (4.3), makes both sequences  $\{f_k\}$  and  $\{F_m\}$  to be periodic. The numerical evaluation of the spectrum  $F(j\omega)$  reduces to evaluating the sequence samples  $\{F_m\}$ .

Since the data observed or analyzed is finite in duration, only a finite number of samples has to be processed. Let there be only  $N$  samples of  $f_k$ . Equation (4.4) reduces to

$$F(e^{j\theta}) = \sum_{k=0}^{N-1} f_k e^{-jk\theta}, \quad (4.5)$$

for the spectrum of the sequence  $\{f_k\}$ . Using the unnormalized frequency variable  $\omega = \theta/T$ , (4.5) can be rewritten as

$$F(e^{j\theta}) = F(e^{j\omega T}) = \sum_{k=0}^{N-1} f_k e^{-j\omega k T}. \quad (4.6)$$

The sequence  $\{f_k\}$  is assumed aperiodic to start with, but  $F(e^{j\omega T})$  is a continuous periodic function. Sampling this spectrum at regular intervals of  $\Delta\omega = \Omega$ , the following equation can be obtained:

$$F_m \triangleq F(e^{jm\Omega T}) = \sum_{k=0}^{N-1} f_k e^{-jkm\Omega T}. \quad (4.7)$$

When  $\Omega = 2\pi/NT$ , there are exactly  $N$  samples [30]  $F_m$  for the spectrum. These samples are periodic since  $F(e^{j\theta})$  is periodic. This choice of  $\Omega$  is useful to precisely invert the discrete Fourier transform of (4.7). The original sequence  $\{f_k\}$  can be obtained from  $\{F_m\}$  using the inverse relation,

$$f_k = \frac{1}{N} \sum_{m=0}^{N-1} F_m e^{jkm\Omega T} . \quad (4.8)$$

Equations (4.7) and (4.8) constitute the discrete Fourier transform (DFT) pair. The transformation is exact, i.e., it is a one to one mapping between  $\{f_k\}$  and  $\{F_m\}$ .

The sampling of the spectrum in (4.7), forces  $\{f_k\}$  to be a periodic sequence outside the  $(0, N-1)$  interval. Initially  $\{f_k\}$  was assumed to be aperiodic. Thus outside  $(0, N-1)$ , (4.8) will be erroneous. To reduce errors, in this numerical evaluation of spectra, either  $N$ , the number of samples, has to be large or the spacing between samples,  $T$ , should be small. Both sequences now have a period of  $N$ . Thus the original time waveform is represented by  $N$  samples spaced  $T$  seconds apart, and the total duration of the waveform is  $R = NT$  seconds;  $\Omega$  is the spacing between the frequency samples and also defines the frequency resolution of the spectrum. Thus, harmonic analysis is performed using the discrete Fourier transform. This is useful for the numerical evaluation of spectra using the speed and efficiency of computers.

### 4.2.2 Spectral Leakage

Equations (4.7) and (4.8) can also be interpreted as an orthogonal projection technique on the time samples. The functions  $e^{-jkm\Omega T}$  are an orthogonal basis set of functions that span the observation interval  $NT$ . Breaking them up into sines and cosines, the following orthogonal functions can be obtained:

$$\begin{aligned}\cos[km \Omega T] &= \cos\left[km \cdot \frac{2\pi}{NT} \cdot T\right] = \cos\left[\frac{2\pi}{NT} km\right] \\ \sin[km \Omega T] &= \sin\left[km \cdot \frac{2\pi}{NT} \cdot T\right] = \sin\left[\frac{2\pi}{NT} km\right]\end{aligned}\tag{4.9}$$

for  $m, k = 0, 1, 2, \dots, N-1$ . These functions give the projections of a given complex sequence along the real and the imaginary axes, respectively.

By orthogonality, it is meant that [33],

$$\sum_{k=0}^{N-1} e^{-jkm\Omega T} e^{jkn\Omega T} = \begin{cases} N & \text{if } m=n \pmod{N} \\ 0 & \text{if } m \neq n \end{cases}\tag{4.10}$$

and

$$\sum_{m=0}^{N-1} e^{-jkm\Omega T} e^{jlm\Omega T} = \begin{cases} N & \text{if } k=l \pmod{N} \\ 0 & \text{if } k \neq l . \end{cases}$$

These functions are thus, orthogonal on finite point sets and are called orthogonal sequences. These sequences are said to span a signal space, into which a given time sequence can be projected, in much the same way as the vector projections in a 3-dimensional space (or the Fourier series of a periodic waveform). Here, it is an  $N$ -dimensional space, since there are  $N$  basis functions.



Using these basis functions, the given sequence is expanded, to find the projection on each one. These projections correspond to the spectral content of the data sequence, since the basis functions represent the frequency sampling points. As long as the given time sequence has a periodicity that matches with one of these basis functions, there is only one spectral component. A periodicity that is slightly away from one of the basis functions (the frequency sampling point or the DFT bin, as it is usually called), projects on all these orthogonal directions and thus biases the estimates of the actual components at these directions. This phenomenon is called spectral leakage, and is due to the finite extent of the data as well as the discretization of the spectrum. This phenomenon is explained by Harris [25].

The reason for this leakage is that a periodicity is being forced on the given finite extent data, and this leads to discontinuity, in the data and its slopes, at the edges of the observation interval. This effect can be reduced by smoothing the data so that it goes to zero at the boundaries. This smoothing is done by a window function (a sequence) that is applied to the input data sequence multiplicatively. In the frequency domain, this leads to a convolution operation and hence, to reduce the bias in the spectral estimation and spectral leakage, a narrow-main-lobe-and-low-side-lobes window function is needed. With such a window function applied to the data, when a periodicity of arbitrary frequency exists in the data sequence, the projection is onto a basis vector closest to this frequency. The projection on the other basis vectors get reduced because of the low side lobes in the window's transform.

### 4.3 Figures of Merit for Windows

Having discussed the utilities and requirements of a window, both in the filter design situation and the harmonic analysis situation, the next discussion is about how to measure their performance quantitatively and how the different data windows, available in literature, can be compared to one another. Harris [25] has listed many parameters that are useful in comparing window performances. He has tabulated the various figures [p. 55], for different windows that were popular at that time [1978]. Some of these figures of merit are described briefly in the following paragraphs.

#### 4.3.1 Equivalent Noise Bandwidth

When a window is used to estimate a harmonic in the presence of broadband noise, the noise within the bandwidth of the window contributes to the estimate of that harmonic, and thus creates a bias in the estimate. To reduce this bias a narrow band window is needed. This bias is quantified by what is known as the equivalent noise bandwidth (ENBW) of the window. This is the width of a rectangle window with the same peak power gain that introduces the same noise power, in the estimation of the harmonic. Here, power implies the squared amplitude of the harmonic. The ENBW is given by

$$ENBW = \frac{\sum_n w_n^2}{\left[ \sum_n w_n \right]^2}, \quad (4.11)$$

where  $\{w_n\}$  are the window coefficients. The summation in the numerator is the



normalized noise power, and that in the denominator is the peak power gain (the gain at zero frequency).

### 4.3.2 Processing Gain and Processing Loss

The DFT can be construed as a bank of matched filters, where each of the filters is matched to one of the sinusoidal basis functions. Thus, a quantity called the coherent gain or processing gain (PG) can be defined for these matched filters, and also for the windows. Since windowing reduces the data to zero at the edges, this can be attributed as the processing loss (PL) of the window. To measure these attributes quantitatively, consider a simple input data sequence of the form

$$\{f_n\} : f_n = A e^{j\omega_k nT} + q_n , \quad (4.12)$$

where it is assumed, for convenience, that the input complex sinusoids have the same frequencies as the basis set, and  $\{q_n\}$  is a white noise sequence that is Gaussian with variance  $\sigma_q^2$ . At the filter output, that is after the window is applied to this sequence, when the DFT output is analyzed, the expected value of the signal at the output (the spectral component at  $\omega_k$ ), is

$$F_k|_{\text{signal}} = \sum_n w_n A e^{j\omega_k nT} e^{-j\omega_k nT} = A \sum_n w_n . \quad (4.13)$$

This is the output of the filter that is 'matched' to the basis frequency  $\omega_k$ . It is found that the output is modified in amplitude by an extra factor which is simply the sum of all the window coefficients. This summation is called the



coherent gain. The square of the this gain is the coherent power gain of the window. For a rectangle window the coherent gain is  $N$ , the number of terms in the window sequence, since each coefficient is unity. For other windows, it is less than  $N$ , since the window coefficients go to zero on either side of the observation interval  $NT$  (about the center of this interval). The coherent gain is normalized by dividing it by  $N$ , and thus the rectangle window, which is used as a reference for the other windows, has unit gain.

The noise component  $q_n$  in (4.12), gives rise to the incoherent component at the output, and the expected noise component at the output is given by

$$F_k|_{\text{noise}} = \sum_n w_n q_n e^{-j\omega_k nT} \quad (4.14)$$

and the square of this expression gives the incoherent power in the output at the DFT sample point  $\omega_k$ . This incoherent power is given by

$$\begin{aligned} E \left\{ |F_k|_{\text{noise}}|^2 \right\} &= \sum_n \sum_m w_n w_m E \{ q_n q_m^* \} e^{-j\omega_k nT} e^{j\omega_k mT} \\ &= \sigma_q^2 \sum_n w_n^2. \end{aligned} \quad (4.15)$$

The summation in the above expression is the incoherent power gain. Thus, the incoherent power gain is the sum of the square of the window coefficients, while the coherent power gain is the square of the sum of the window coefficients.

The processing gain, PG, is defined as the ratio of the output SNR (signal-to-noise-ratio) to the input SNR, and is hence given by

$$PG = \frac{A^2 \left[ \sum_n w_n \right]^2 / \sigma_q^2 \sum_n w_n^2}{A^2 / \sigma_q^2} = \frac{\left[ \sum_n w_n \right]^2}{\sum_n w_n^2} . \quad (4.16)$$

The reciprocal of this gives the processing loss, which is the same as the ENBW defined in (4.11).

#### 4.3.3 Scalloping Loss

This is also called the picket-fence effect. In (4.16), the processing gain was considered when the input tone lies at an exact DFT sampling point. There would be a reduction in PG, as an input tone resides in between neighboring basis vectors, and this reduction is maximum when the tone is at the center of the two basis vectors. Now (4.13), which gives the coherent signal component at the output, becomes

$$F_k |_{\text{signal}} = \sum_n w_n A e^{j\omega_{(k+1/2)} nT} e^{-j\omega_k nT} , \quad (4.17)$$

where  $\omega_{(k+1/2)}$  is the midpoint between  $\omega_k$  and  $\omega_{(k+1)}$ , and it is, hence, given by

$$\omega_{(k+1/2)} = 2\pi (k+1/2) \frac{f_s}{N} = 2\pi (k+1/2) \frac{1}{NT} . \quad (4.18)$$

Putting (4.18) into (4.17), we have

$$F_k |_{\text{signal}} = A \sum_n w_n e^{j(\pi/N)n} . \quad (4.19)$$

The scalloping loss is defined as the ratio of the coherent gain for a tone at the midpoint between two DFT bins, to the coherent gain for a tone at a DFT bin. Hence scalloping loss is expressed as

$$\text{Scalloping loss} = \frac{\left| \sum_n w_n e^{j(\pi/N)n} \right|}{\sum_n w_n} \quad (4.20)$$

The power loss is the square of the above expression, and is usually expressed in decibels.

The sum of the two processing losses, viz., the maximum scalloping loss due to the worst case frequency location of the tone and the processing loss due to data reduction at the edges of the observation interval, is called the worst case processing loss. This loss is also, usually, expressed in decibels, and is the one mostly listed in the table of merits.

#### 4.3.4 Minimum Resolution Bandwidth

As long as one is just interested in detecting a harmonic in broad-band noise, any window other than the rectangle window, is useful. But when one has to resolve two nearby tones of nearly equal strength, one needs windows with special characteristics. More than a minimum worst case processing loss, is needed for the window. The window should have a narrow main lobe and a low



side lobe structure, to resolve the two nearby tones. A quantity that is used in measuring a window performance, in relation to this resolving problem, is the minimum resolution bandwidth. This is the minimum separation required between two equal-amplitude tones at the input, that is necessary to resolve them at the output of the windowed transform. The usual specification is the half-power or 3 dB bandwidth, that measures the bandwidth of a filter. In using the DFT, there is a coherent addition of the spectral components, weighted through the window coefficients, at a given frequency. The half-power specification is not sufficient in this case. When two kernels are contributing to the coherent summation, the sum at the crossover point should be less than the amplitude of the tones, if they have to be resolved. Hence the gain from each kernel, at the crossover point should be less than half, or the crossover point should be beyond the 3 dB point. It is usual to consider the 6 dB bandwidth for the purposes of resolution. The two kernels arise in this situation due to the window being centered at each of the input tones, when one measures the spectrum at the crossover point. The two equal-amplitude tones contribute to the output at the DFT bins in the crossover region. The 6 dB bandwidth is taken to be the minimum resolution bandwidth of the window.

#### **4.3.5 Summary and Other Merits**

Besides the above figures of merit, a few others can be found in [25]. But the above quantities are very useful in describing windows as applied to harmonic

analysis. It is usual to compare and contrast different windows based on

- (i) the side lobe level, which reduces spectral leakage, and hence, the estimation bias,
- (ii) the worst case processing loss, which affects detectability of a tone, and
- (iii) the minimum resolution bandwidth, which specifies the spacing necessary between two equal-amplitude tones, for being resolved.

Also a good window should have a side lobe structure that decays very fast with frequency, for this reduces interference from distant tones. The computation of the window coefficients should be easy, and quantizing the coefficients should not adversely affect the window performance.

#### 4.4 The Prolate Window

The DPSS  $\{\nu_n^{(0)}(N, W)\}$ , on the index set  $(0, N-1)$ , constitutes the window coefficients and hence this window is called the prolate window. A small  $W$  can give the required narrow main lobe. Thus the DPSS lends itself to a flexible window design. This window is compared to other standard and popular windows, in the following sections.

##### 4.4.1 Comparison to Other Windows

There are numerous window designs available in literature. Two of the fundamental ones are the rectangle and triangle windows. Though not used in practice, these form a basis of comparison for other windows. The Hamming and Hanning windows, that were obtained by adding translated Dirichlet kernels [25], are simple to compute and have a good resolving power and a low processing loss.



The Blackman-Harris family of windows evolve as a 3-term or 4-term weighted cosine series. The weighting coefficients can be chosen, to place the zeros in the frequency response at required locations or to minimize the side lobe levels. This family of windows is highly useful for harmonic analysis, since it has good resolving power, low side lobes and the window coefficients are easy to compute. The Dolph-Chebyshev family of windows have a very narrow main lobe for any given side lobe level (all the side lobes are at the same height). It is not very easy to compute this window, and the side lobe levels are very sensitive to inaccuracies in the computation of the window coefficients. Other optimal windows such as the Kaiser-Bessel and Barcilon-Temes windows also perform as well as the Blackman-Harris windows, in detecting a weak signal in the presence of a strong nearby one.

Table 1 lists the figures of merit, that were discussed in the previous section, of a few of these windows. The prolate window is computed for the values of the parameter  $W = 0.03, 0.05, 0.06$  and  $0.08$ . From Table 1, it is seen that the windows have a worst case loss in the range of 3 to 4 dB, and a minimum resolution bandwidth in the range of 1.2 to 2.8 bins. Of the two 4-term windows, the 'minimum' 4-term Blackman-Harris window has a highest side lobe at  $-92.5$  dB (Nuttall [34], came up with a 4-term window that had a  $-93.32$  dB high side lobe. This level is less than the one Blackman-Harris claimed to be minimum, and hence Nuttall phrases that window as 'minimum'). The other 4-term Blackman-Harris window has a  $-74$  dB high side lobe. For convenience, the Blackman-Harris



**Table 1 – Some Figures of Merit of Selected Windows**

WINDOW	Highest Side Lobe Level (dB)	Coherent Gain	ENBW (Bins)	3 dB Band-Width (Bins)	Scalloping Loss (dB)	Worst Case Loss (dB)	6 dB Band-Width (Bins)
Rectangle †	– 13.3	1.00	1.00	0.89	3.92	3.92	1.21
Triangle †	– 26.6	0.50	1.33	1.27	1.82	3.07	1.77
'Minimum' 4-Sample † Blackman-Harris	– 92.5	0.36	2.00	1.90	0.83	3.84	2.66
74 dB 4-Sample † Blackman-Harris	– 74.8	0.40	1.79	1.70	1.03	3.56	2.38
Prolate ( $W = 0.03$ )	– 33.1	0.58	1.29	1.24	1.93	3.04	1.72
Prolate ( $W = 0.05$ )	– 57.2	0.45	1.62	1.54	1.24	3.35	2.16
Prolate ( $W = 0.06$ )	– 70.0	0.40	1.77	1.68	1.06	3.53	2.35
Prolate ( $W = 0.08$ )	– 91.8	0.35	2.06	1.95	0.78	3.93	2.74
Dolph-Chebyshev † $\alpha = 2.5$	– 50.0	0.53	1.39	1.33	1.70	3.12	1.85
Dolph-Chebyshev † $\alpha = 3.0$	– 60.0	0.48	1.51	1.44	1.44	3.23	2.01
Dolph-Chebyshev † $\alpha = 3.5$	– 70.0	0.45	1.62	1.55	1.25	3.35	2.17
Dolph-Chebyshev † $\alpha = 4.0$	– 80.0	0.42	1.73	1.65	1.10	3.48	2.31

† Refer [25], page 55, for more information on these windows.

windows are addressed by their side lobe levels. Of the two 4-term Blackman-Harris windows, except for its inferior high side lobe, the 74-dB window has a lower 6 dB bandwidth, hence a higher resolving power; a higher coherent gain, hence a lower ENBW; a slightly higher scalloping loss, but a lower worst case processing loss.

Of the four prolate windows, the one with  $W = 0.06$ , has an overall better performance, in terms of its side lobe level, scalloping loss, coherent gain and worst case processing loss. Comparing this window to the 74-dB Blackman-Harris window, it is observed that the prolate window has a slightly lower 6 dB bandwidth and hence a slightly better resolving power; a lower worst case processing loss; a higher first side lobe level at  $-71$  dB but a faster side lobe falloff. These differences are, however, not very significant, and the windows are very similar.

#### 4.4.2 Side Lobe Level vs. Worst Case Processing Loss

The side lobe level and the worst case processing loss are, perhaps, the two most important parameters of the windows, in relation to harmonic analysis. Harris [25], accordingly compared several windows by locating them in a 2-dimensional plane, that has the side lobe levels marked on the  $y$ -axis and the worst case processing loss on the  $x$ -axis. The windows located on the lowest left-hand corner, would be the best ones for harmonic analysis, since they have both, a low side lobe level and a low worst case processing loss. The rectangle, triangle, Dolph-Chebyshev, Blackman-Harris and prolate windows, are located on this graph (see Fig. 12). The Dolph-Chebyshev windows are located at the lowest left-hand region in this plane, and hence they are the best windows for harmonic analysis. But their side lobe structure is very sensitive to errors in the computation of the window coefficients, and hence their location on this graph is prone to change. Moreover, all the side lobes are at the same height and this is not suitable for multi-tone detection since even a distant strong tone creates the same amount of bias, in the estimation of a weak tone, as does a nearby strong tone.



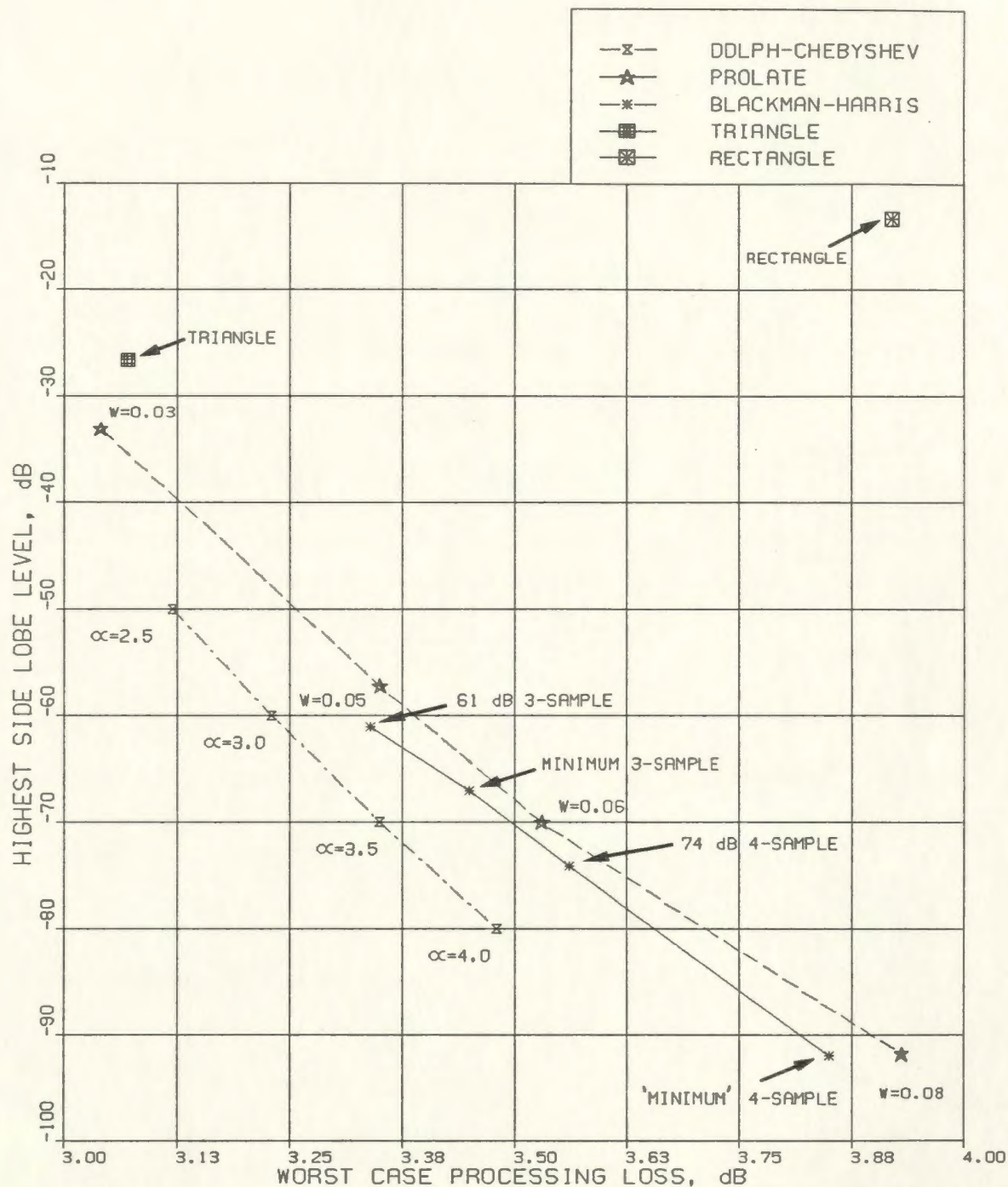


FIG. 12 - COMPARISON OF WINDOWS BASED ON SIDE LOBE LEVEL AND WORST CASE PROCESSING LOSS



The prolate windows and the Blackman-Harris windows (so also the Kaiser-Bessel windows) are next-in-line, with respect to their location on this plane. They do not have the constant level side lobes and coefficient sensitivity problems of the Dolph-Chebyshev windows. Hence the prolate, Blackman-Harris and Kaiser-Bessel windows are a better choice for harmonic analysis. Though useful for comparing windows, this 2-dimensional plane does not fully represent the window performances. In the following section the Blackman-Harris and prolate windows are compared in the situation of detecting a weak tone in the vicinity of a strong tone, based on an example in Harris [25].

#### 4.5 The Two Tone Detection Problem

A problem of two tone detection, that proves the usefulness and necessity of a data window, is now considered. The prolate window that is considered here is the one with  $W = 0.06$  and the Blackman-Harris window is the one with a  $-74$  dB high side lobe. The rectangle window's performance is also included in this situation, to highlight its deficiency. The frequency spectrum of the three windows are provided in Fig. 13 & 14. The rectangle window is seen to have a very narrow (like a pencil-beam) main lobe, but the side lobe levels are very high, the highest side lobe is at a height of about  $-13$  dB. The prolate and Blackman-Harris windows have very low level side lobes, and a main lobe much wider than that of the rectangle window. It is also observed that the side lobe falloff is faster for the prolate window, as said earlier. This makes them more suited to multi tone detection situations, as a distant tone would cause lesser interference in comparison to a nearby tone.

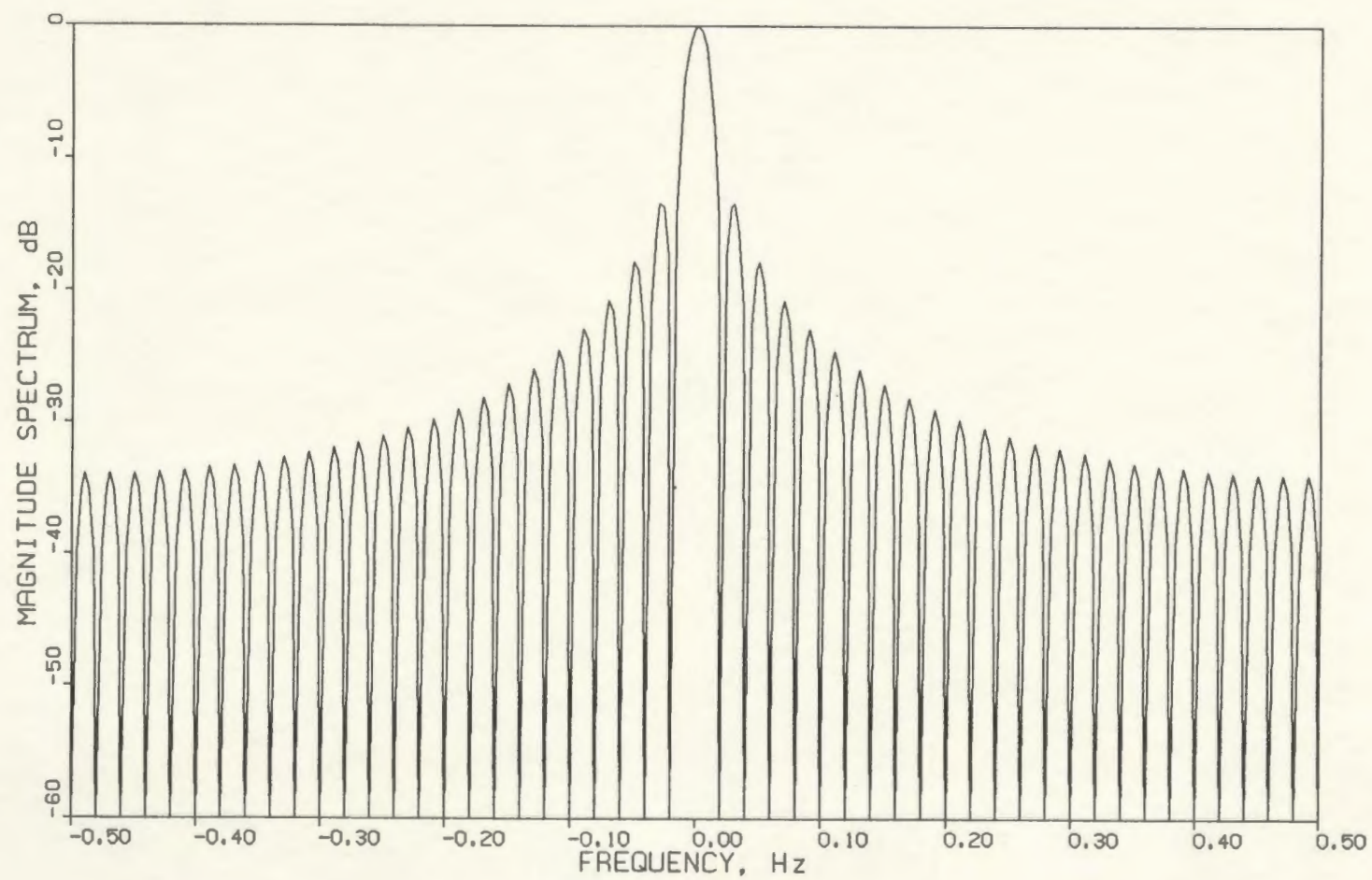


FIG. 13 - RECTANGLE WINDOW  
FREQUENCY RESPONSE OF THE WINDOW

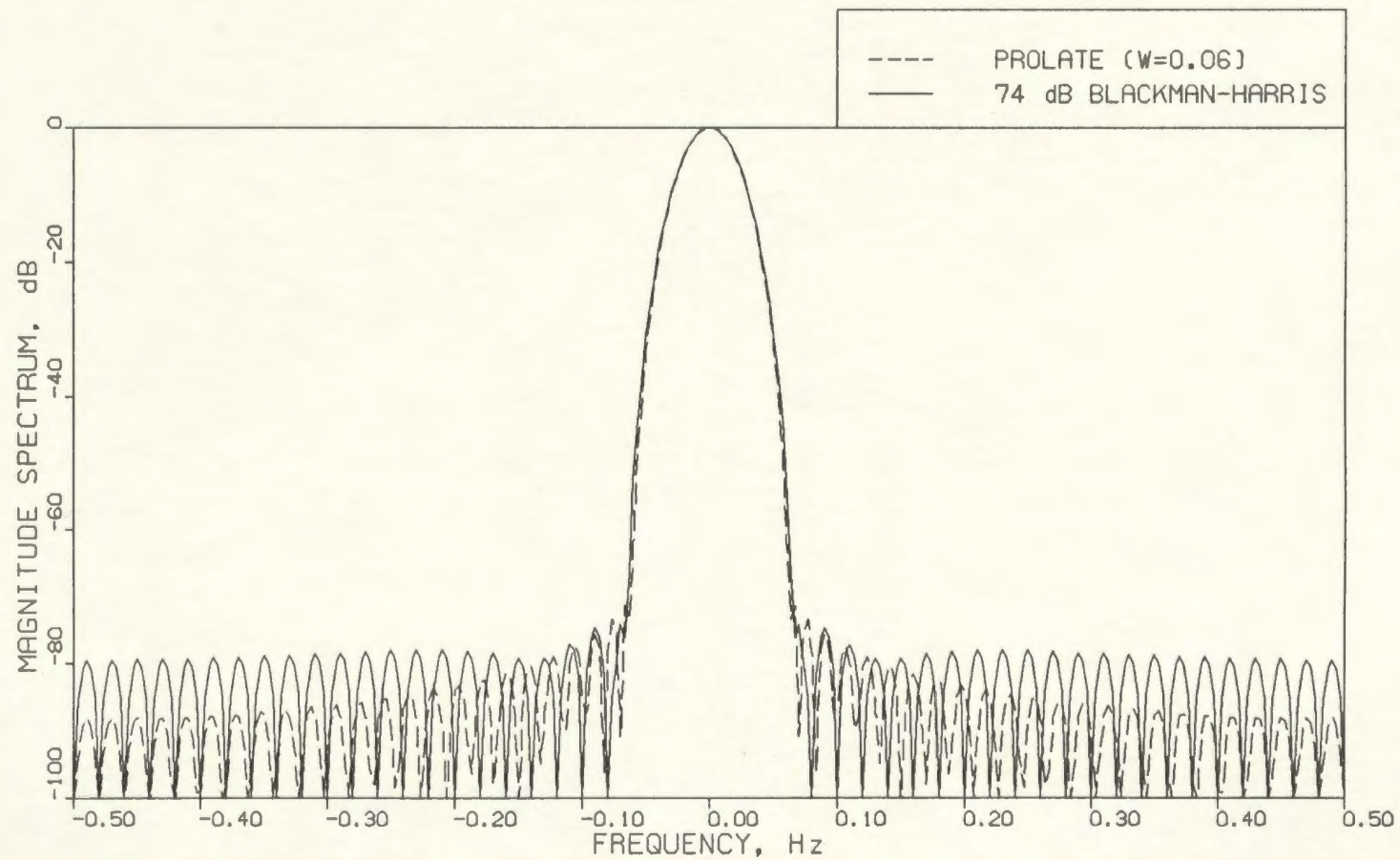


FIG. 14 - BLACKMAN-HARRIS & PROLATE WINDOWS  
COMPARISON OF FREQUENCY RESPONSE



#### 4.5.1 The Tones at 5.0 and 10.0 Bins

When two spectral lines reside in DFT bins, the rectangle window allows each to be identified with no interaction. To demonstrate this, the two input tones are chosen to be at  $5.0 f_g/N$  and  $10.0 f_g/N$ , corresponding to the 5th and 10th DFT bins. The tone at the 5th bin is of unit amplitude, while the weaker tone at the 10th bin is of 0.01 amplitude. Thus, on a log scale, there is a 40 dB separation in their amplitudes. The power spectrum at the rectangle window output is shown in Fig. 15. The DFT output points are joined by straight line segments. Clearly, two peaks are visible at the output, showing that the rectangle window has not caused any interference of the two tones. The DFT bins correspond to the nulls of its spectrum, and thus no leakage (and hence no bias) is caused in the estimation of the tones.

For the same input, the prolate and the Blackman-Harris windows give a rather broad spectrum DFT output. The DFT outputs are shown in Fig. 15. But two peaks are distinct, and these correspond to the two input tones; there is a difference in the heights of the peaks of 40 dB magnitude, and this is the same as that of the input tones. For the prolate window there is a pronounced null between the peaks, that is 25 dB below the smaller peak. The null in the case of the Blackman-Harris window is 20 dB below the smaller peak. If a 3 dB margin is considered for high confidence detection, then the detection is good with both these windows.

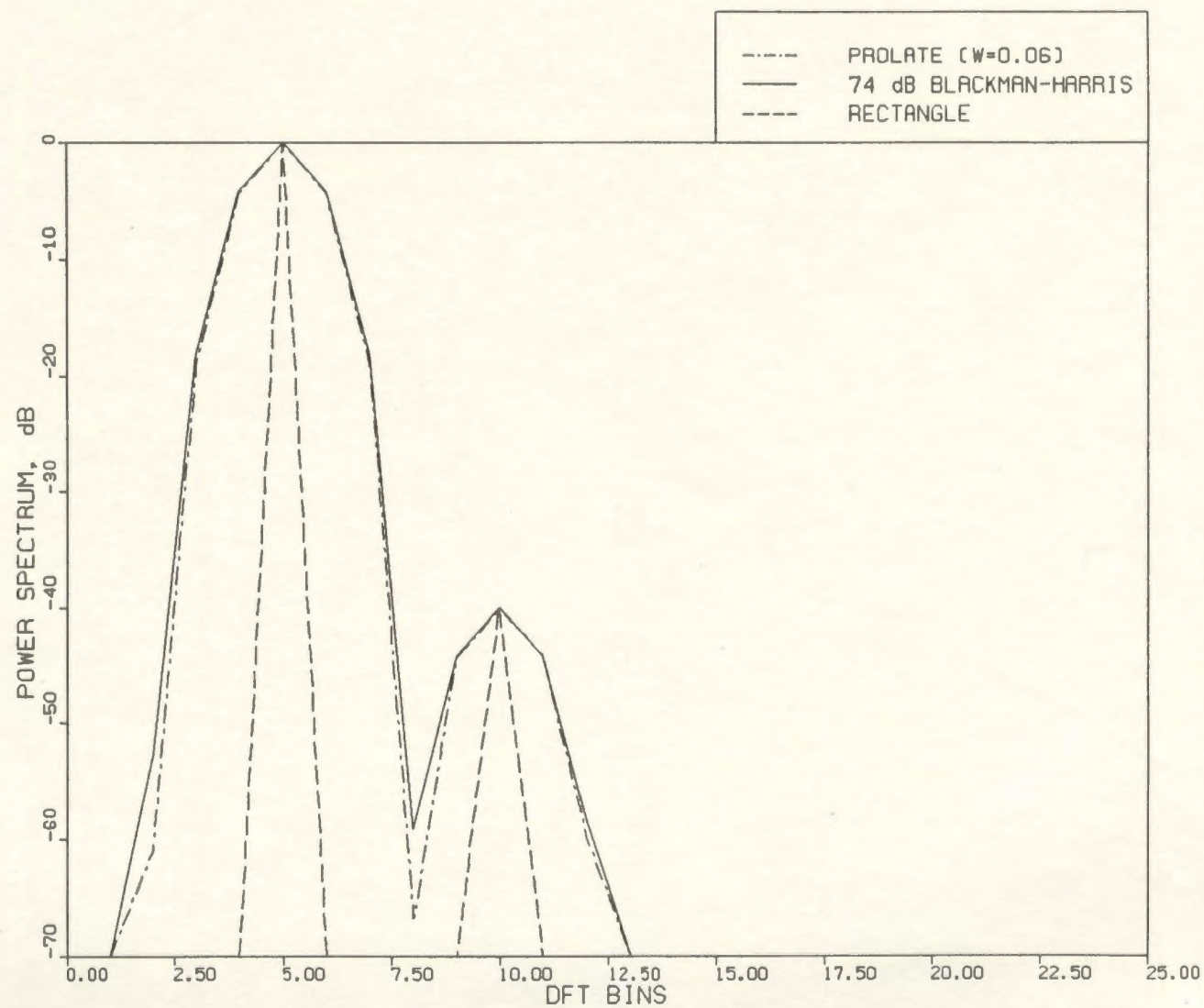


FIG. 15 - INPUT TONES AT 5.00 & 10.00 BINS  
POWER SPECTRUM AFTER WINDOWING



### 4.5.2 The Tones at 5.5 and 10.0 Bins

The input is now modified, so that the larger tone is shifted to 5.5 bins (a location corresponding to the worst case processing loss, for most of the windows), and the smaller tone is stationed at 10.0 bins. The results can be seen in Fig. 16. The rectangle window just detects one peak corresponding to the larger tone; there is just one peak in the DFT output of the rectangle window. The side lobe structure of the larger tone engulfs completely the main lobe of the smaller tone. The separation between the input tones is now 4.5 bins. At this frequency, the rectangle window has a gain of  $-23$  dB with respect to the d.c. gain (see Fig. 13). The window has a worst case processing loss of 4 dB, and considering a 3.0 dB high confidence detection margin, the 'threshold' level at this 4.5 bins separation is  $-23 - 4 + 3 = -24$  dB. The smaller tone at  $-40$  dB, is thus within this 'threshold' and thus it is engulfed in the side lobe structure of the larger tone. There is also an asymmetry noticed in this side lobe structure, about the main lobe at 5.5 bins. This is due to the coherent addition of the side lobe structures (discussed earlier in the section on spectral leakage) between the kernels located at the positive and negative frequency axes. There is a leakage between these two kernels.

The prolate and Blackman-Harris windows are now applied to the two tone signal, to illustrate the weak tone detectability (Fig. 16). The windows have very similar DFT outputs. There are two peaks with 40 dB magnitude separation with a 4 dB null separating them. For the same 3 dB high confidence margin, the



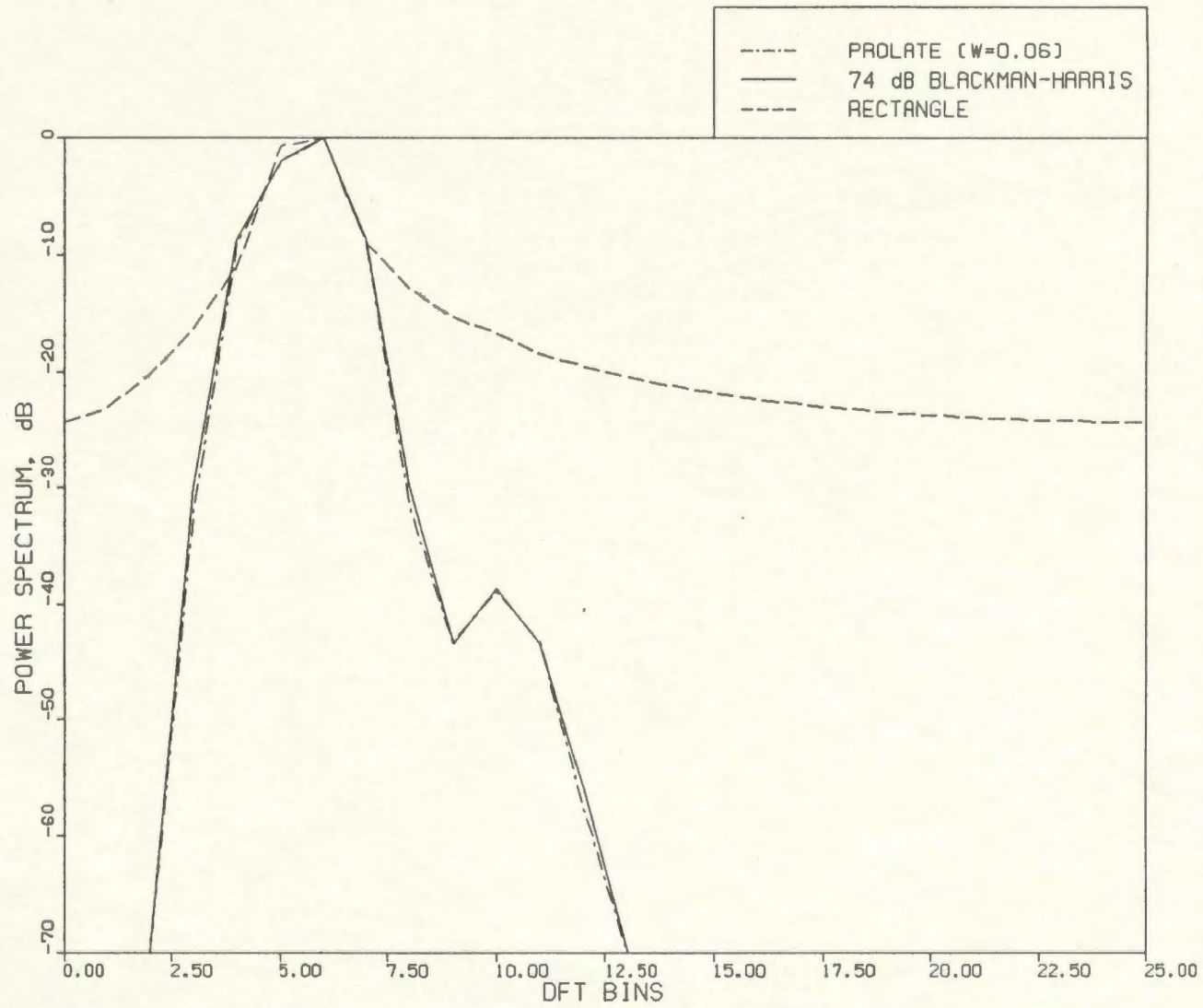


FIG. 16 - INPUT TONES AT 5.50 & 10.00 BINS  
POWER SPECTRUM AFTER WINDOWING

detection is thus marginal. Also, since there is no DFT output corresponding to 5.5 bins, the larger tone seems to be located at 6.0 bins and thus there is a slight positional bias in the estimation. This experiment has demonstrated the limitations of the rectangle window, which detects only one peak, and the usefulness of the other two windows.

#### **4.5.3 The Tones at 5.25 and 10.0 Bins**

The input power spectrum is modified so that the larger tone is shifted to 5.25 bins. Then it is checked if this corresponds to the worst case location for any of the three windows being compared here. The weaker tone is still positioned at 10.0 bins. The rectangle window again detects only one peak (that of the larger tone), and the result is similar to the previous case (Fig. 17), except that the asymmetry about the main lobe is more pronounced and there is a reduced side lobe level. The weaker tone is still within the side lobe structure of the larger tone and hence goes undetected.

Again, the Blackman-Harris and prolate windows are tried. The results are quite similar to the previous case, viz., two output peaks, with a 40 dB separation in their heights and at the input frequency locations. The null between peaks, for the Blackman-Harris window, is 5 dB deep, and that for the prolate window is 2 dB better. The detection is slightly better than in the previous case. This confirms that the previous case, viz., larger tone at 5.5 bins, corresponds to the worst case location for all the three windows.

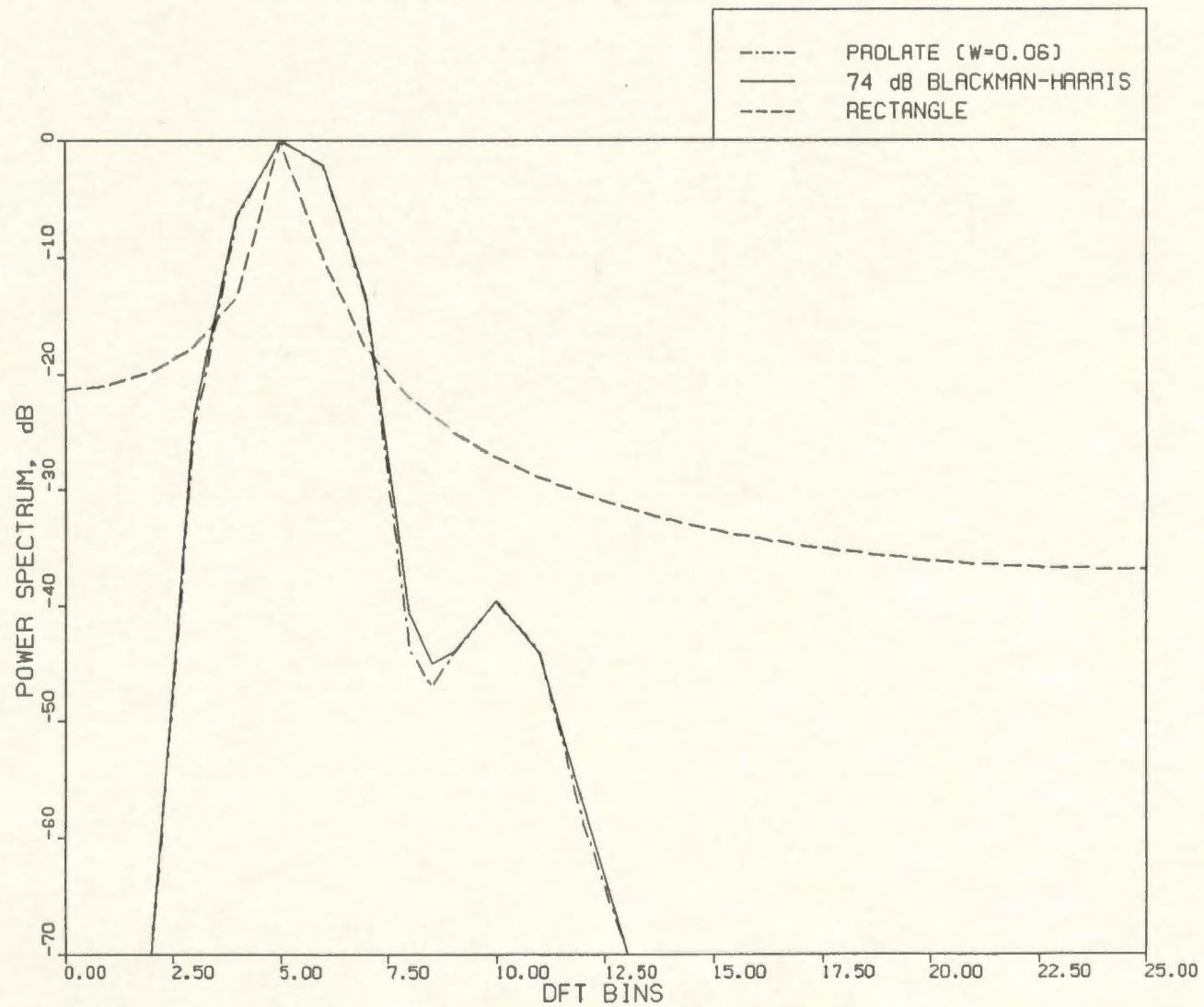


FIG. 17 - INPUT TONES AT 5.25 & 10.00 BINS  
POWER SPECTRUM AFTER WINDOWING



The prolate window performs as well as the Blackman-Harris window, as seen in this example. Also the Kaiser-Bessel, Dolph-Chebyshev and Barcilon-Temes windows are good for this situation, as Harris [25] demonstrated. The main lobe width of the prolate window can be controlled by varying the parameter  $W$  of the DPSS  $\{v_n^{(0)}(N, W)\}$ . Thus the main lobe width can be smoothly controlled. This flexibility does not exist in the design of the Blackman-Harris windows. For larger  $W$ , the prolate window has lower side lobe levels. A gradient search technique is used, in the case of the Blackman-Harris window, to find the weighting coefficients in the cosine series that achieve a minimum side lobe level.

## CHAPTER 5

### SUMMARY AND CONCLUSIONS

This study is basically twofold:

- (i) to investigate the use of higher order discrete prolate functions in filter design, and
- (ii) to apply the prolate function, as a window, in spectral analysis.

The passband performance has been improved, in the present work. The use of the prolate function in harmonic analysis has been demonstrated.

The discrete prolate filter is based on the DPSWF  $U_0(N, W; f)$ , as seen earlier. When  $U_2(N, W; f)$ , the next higher order even function, is added to  $U_0(N, W; f)$ , a widened main lobe results. The main lobe is also less slanting, and the edge-of-the-passband gain is also improved. As more and more even functions are added there is an increase in main lobe width and an improvement in flatness. The side lobe falloff is also faster. These improvements level off after the first few additions, since the passband energy concentration of higher order functions becomes less and less significant with increasing order.

However, the addition of higher order terms also introduces passband ripples, as  $N$  and  $W$  get larger. This is an unwanted situation, but the ripple amplitudes reduce with the number of even functions, with significant passband energy concentration, that are included in the design. Also, the side lobe levels of

the prolate filter increase with the inclusion of the higher order functions. This is expected, since it is known that the higher order functions have higher energy concentrations outside the passband interval, as compared to that of the lowest order one. This increase in side lobe levels can be reduced to some extent with the inclusion of all the significant orders.

One of the filters, so designed by including higher order DPSWF, is compared to a Chebyshev filter with similar design parameters. It is seen that the new design is as good as the Chebyshev design, and would thus be suitable to many applications that warrant a Chebyshev type filter. Thus, it is seen that the passband performance of the prolate filter can be improved at the cost of raised side lobe levels. There is a reduction of energy concentration in the passband, in including higher order DPSWF; but the passband energy concentration is comparable to that of other filters with a flat passband.

The addition of the prolate functions is performed with a least-squares criterion, in choosing the weights for the higher order functions. A possible extension to this work, would be to use a Chebyshev criterion in choosing these weighting coefficients to achieve a flat passband characteristic.

In the second part of this work, the narrow main lobe and the low side lobe structure of the DPSWF  $U_0(N, W; f)$  was made use of, in harmonic analysis. For this purpose, the DPSS  $\{v_n^{(0)}(N, W)\}$  on the index set  $(0, N-1)$  was used as the window coefficients, in analyzing the harmonic contents of a data sequence



using the discrete Fourier transform. This window is appropriately termed as the prolate window. The width of the main lobe can be controlled by a proper choice of  $W$ , and this leads to flexible window designs.

With reference to detection of a harmonic tone in the presence of strong interference, the windows that are best suited, are the Kaiser-Bessel and the Blackman-Harris families of windows. The performance of the prolate window was compared to that of the 4-term Blackman-Harris window. The windows are located in the 2-dimensional plane (see Fig. 12), that compares the windows on the basis of their highest side lobe level and their minimum resolution bandwidth. The prolate windows are located close to the Blackman-Harris windows, and the windows have a similar performance in the two tone detection problem that was considered.

Thus it is found that the prolate window is as useful to applications in spectral analysis, as the popular Blackman-Harris and Kaiser-Bessel windows. Another application of this type is picture processing. It would be interesting, to extend the prolate window to the two-dimensional case and use it for edge detection in pictures.

## REFERENCES

1. Sinha, B.P. and MacPhie, R.H., 'Electromagnetic Scattering from Prolate Spheroids for Axial Incidence,' IEEE Transactions on Antennas and Propagation, **AP-23**, 1975, pp. 676-679.
2. Sinha, B.P. and MacPhie, R.H., 'Electromagnetic Scattering by Prolate Spheroids for Plane Waves with Arbitrary Polarization and Angle of Incidence,' Radio Science, Vol. **12**, No. 2, 1977, pp. 171-184.
3. Dalmas, J. and Deleuil, R., 'Scattering of Electromagnetic Waves by a Prolate Spheroid and a Half Prolate Spheroid Lying on a Plane with Axial Incidence,' Optica Acta, Vol. **27**, No. 5, 1980, pp. 637-649.
4. Dalmas, J., 'Scattering of Electromagnetic Waves by an Ellipsoid with an Elongated Revolution of Infinite Conduction at Non-axial Incidence,' Optica Acta, Vol. **28**, No. 7, 1981, pp. 933-948.
5. Sebak, A.A. and Shafai, L., 'Scattering by Imperfectly Conducting and Impedance Spheroids,' Radio Science, Vol. **19**, No. 1, 1984, pp. 258-266.
6. Sinha, B.P. and Cooray, M.F.R., 'Electromagnetic Scattering by Dielectric Prolate Spheroids,' URSI Meeting, Boulder, Colorado, Jan. 13-16, 1986.
7. Slepian, D. and Pollak, H.O., 'Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - I,' Bell System Technical Journal, **40**, 1961, pp. 43-64.
8. Landau, H.J. and Pollak, H.O., 'Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - II,' Bell System Technical Journal, **40**, 1961, pp. 65-84.
9. Landau, H.J. and Pollak, H.O., 'Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - III: The Dimension of the space of Essentially Time- and Band-limited signals,' Bell System Technical Journal, **41**, 1962, pp. 1295-1336.
10. Slepian, D., 'Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - IV,' Bell System Technical Journal, **43**, 1964, pp. 3009-3058.
11. Slepian, D., 'Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - V: The Discrete Case,' Bell System Technical Journal, **57**, 1978, pp. 1371-1430.



12. Tufts, D.W. and Francis, J.T., 'Designing Digital Low-pass Filters — Comparison of some Methods and Criteria,' IEEE Transactions on Audio and Electroacoustics, **AU-18**, No. 4, 1970, pp. 487-494.
13. Papoulis, A. and Bertran, M.S., 'Digital Filtering and Prolate Functions,' IEEE Transactions on Circuit Theory, **CT-19**, No. 6, 1972, pp. 674-681.
14. Eberhard, A., 'An optimum Discrete Window for the Calculation of Power Spectra,' IEEE Transactions on Audio and Electroacoustics, **AU-21**, No. 1, 1973, pp. 37-43.
15. Jain, A.K. and Ranganath, S., 'Extrapolation Algorithms for Discrete signals with Application in Spectral Estimation,' IEEE Transactions on Acoustics, Speech, and Signal Processing, **ASSP-29**, No. 4, 1981, pp. 830-845.
16. Dickey, F.M. and Sam Shanmugam, K., 'Optimum Edge Detection Filter,' Applied Optics, **16**, No. 1, 1977, pp.145-148.
17. Sam Shanmugam, K., Dickey, F.M. and Green, J.A., 'An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures,' IEEE Transactions on Pattern Analysis and Machine Intelligence, **PAMI-1**, No. 1, 1979, pp. 37-49.
18. Durrani, T.S. and Chapman, R., 'Constrained Optimization solutions to I.I.R. filter design using discrete prolate spheroidal wave functions,' Proceedings ICASSP, Int Conf Acoust Speech Signal Processing, **1**, 1981, pp. 228-231.
19. Durrani, T.S. and Chapman, R., 'Optimal All-Pole Filter Design Based on Discrete Prolate Spheroidal Sequences,' IEEE Transactions on Acoustics, Speech, and Signal Processing, **ASSP-32**, No. 4, 1984, pp. 716-721.
20. Flammer, C., Spheroidal Wave Functions , Stanford University Press, Stanford, California, 1957.
21. Stratton, J.A., Morse, P.M., Chu, L.J., Little, J.D.C. and Corbató , F.J., Spheroidal Wave Functions , The Technology Press of M.I.T., Cambridge, Mass., and John Wiley and Sons, New York, 1956.
22. Morse, P.M. and Feshbach, H., Methods of Theoretical Physics , McGraw-Hill, New York, 1953.
23. Meixner, J. and Schä fke, F.W., Mathieusche Funktionen und Sphä roid-funktionen , Julius Springer, Berlin, 1954.



24. Sinha, B.P. and MacPhie, R.H., 'On the Computation of the Prolate Spheroidal Radial Functions of the Second Kind,' Journal of mathematical physics, Vol. **16**, No. 12, 1975, pp. 2378-2381.
25. Harris, F.J., 'On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform,' Proceedings of the IEEE, **66**, No. 1, 1978, pp. 51-83.
26. Baker, C.T.H., The Numerical Treatment of Integral Equations , Oxford University Press, Oxford, 1977.
27. James, M.L., Smith, G.M. and Wolford, J.C., Applied Numerical Methods for Digital Computation , Harper & Row, Publishers, New York, 1977.
28. Bowdler, H., Martin, R.S., Reinsch, C. and Wilkinson, J.H., 'The QR and QL Algorithms for Symmetric Matrices,' Numerische Mathematik, **11**, 1968, pp. 293-306.
29. Smith, B.T., Boyle, J.M., Garbow, B.S., Ikebe, Y., Klema, V.C. and Moler, C.B., Matrix Eigensystem Routines-Eispack Guide , Springer-Verlag, Heidelberg, 1974.
30. Gabel, R.A. and Roberts, R.A., Signals and Linear Systems , John Wiley and Sons, Inc., New York, 1980.
31. Rice, J.R., The Approximation of Functions. Linear Theory , Vol. 1, Addison-Wesley Publishing Company, Massachusetts, 1964.
32. Rabiner, L.R. and Gold, B., Theory and Application of Digital Signal Processing , Prentice-Hall, Inc., New Jersey, 1975.
33. Cooley, J.W., Lewis, P.A.W. and Welch, P.D., 'The Finite Fourier Transform,' IEEE Transactions on Audio and Electroacoustics, **AU-17**, No. 2, 1969, pp. 77-85.
34. Nuttall, A.H., 'Some Windows with Very Good Side lobe Behavior,' IEEE Transactions on Acoustics, Speech and Signal Processing, **ASSP-29**, No. 1, 1981, pp. 84-91.

**APPENDIX**  
**PROGRAM LISTINGS**



```

C*****
C*
C*   THIS PROGRAM IS FOR COMPUTING THE DISCRETE PROLATE
C*   SPHEROIDAL SEQUENCES AND THE RELATED DISCRETE
C*   PROLATE SPHEROIDAL WAVE FUNCTIONS (DPSS & DPSWF),
C*   AND FOR DOING THE FOURIER ADDITION OF THE DPSWF
C*   (USING EISPACK ROUTINES)
C*
C*****
C*               LIST OF VARIABLES
C*****
C*
C*   A           = MATRIX WHOSE EIGENVECTORS ARE THE DPSS
C*   N           = ORDER OF A
C*   WR,WI       = VECTORS CONTAINING EIGENVALUES OF A
C*   Z(I,L)      = THE N COMPONENTS OF THE Lth EIGENVECTOR
C*   U(L,J)      = THE DPSWF CORRESPONDING TO Lth EIGENVECTOR
C*   U1(L,J)     = REAL U(L,J)
C*   U2(L,J)     = MAGNITUDE OF U(L,J)
C*   NM          = NUMBER OF ROWS IN Z MATRIX DIMENSION
C*   IERR        = ERROR PARAMETER
C*   ITS         = NUMBER OF ITERATIONS
C*   W           = NORMALISED BANDWIDTH (BETWEEN 0. AND .5)
C*   N1          = NUMBER OF POINTS FOR FREQUENCY PLOT
C*               (NUMBER OF STRIPS FOR INTEGRATION)
C*   N2          = THE INTEGRAL MULTIPLE FACTOR FOR W
C*   DELF        = WIDTH OF THE INTEGRATION STRIP
C*   FREQ        = NORMALISED FREQUENCY (BETWEEN 0. AND .5)
C*   MAG         = MAGNITUDE RESPONSE OF THE FILTER
C*   NS          = NUMBER OF EIGENVALUES USED IN MAG-FREQ PLOT
C*   FLTER1      = THE FOURIER WEIGHTED TRANSFER FUNCTION
C*   FLTER2      = THE LOGARITHMIC FLTER1
C*   COEFF       = THE FOURIER COEFFICIENTS
C*
C*****
C*   SECTION FOR TYPE DECLARATION AND DIMENSIONING
C*****
CHARACTER*1 III,JJJ,KKK,LLL
PARAMETER (N=10,N1=80)
INTEGER ITS(N)
REAL*8 Z(N,N),WR(N),WI(N),A(N,N),FV1(N),IV1(N)
COMPLEX*16 U(N,N1+1),E
REAL*8 U1(N,N1+1),U2(N1+1),SUM1(N),SUM2(N)
REAL*8 FLTER1(N,N1+1),FLTER2(N,N1+1)
REAL*8 COEFF(N),ENERGY(N1+1)

C*****
C*   THE PARAMETERS ARE ENTERED HERE
C*****

NM=N
NS=N
W=.20
PI=ASIN(1.)*2
N2=IFIX(W*N1)
DELF=1./N1

```



```

C*****
C*
C*          MATRIX ELEMENTS ARE DEFINED HERE
C*
C*****

```

```

      DO 20 I=1,N
      DO 15 J=1,N
      IF (I.EQ.J) GO TO 5
      GO TO 10
5      A(I,J)=2*W
      GO TO 15
10     A(I,J)=SIN(PI*2*W*(I-J))/(PI*(I-J))
15     CONTINUE
20     CONTINUE

```

```

C*****
C*
C*          SECTION FOR COMPUTING EIGENVALUES AND EIGENVECTORS
C*
C*****

```

```

      CALL TRED2(NM,N,A,WR,FV1,Z)
      CALL TQL2(NM,N,WR,FV1,Z,IERR,ITS)

```

```

C*****
C*
C*          SECTION FOR MAKING  SUM Z(I,J) > 0,  AND
C*          SUM (N+1-2*I)*Z(I,J) > 0
C*
C*****

```

```

      DO 40 J=1,N
      SUM1(J)=0.
      SUM2(J)=0.
      DO 25 I=1,N
      SUM1(J)=SUM1(J)+Z(I,J)
25     SUM2(J)=SUM2(J)+(N+1-2*I)*Z(I,J)
      IF (SUM1(J).LT.0.) THEN
      DO 30 I=1,N
30     Z(I,J)=-Z(I,J)
      SUM1(J)=-SUM1(J)
      END IF
      IF (SUM2(J).LT.0.) THEN
      DO 35 I=1,N/2
      TEMP=Z(I,J)
      Z(I,J)=Z(N+1-I,J)
35     Z(N+1-I,J)=TEMP
      SUM2(J)=-SUM2(J)
      END IF
40     CONTINUE

```

```

C*****
C*
C*      SECTION FOR COMPUTING THE DPSWF FROM THE DPSS
C*
C*****

```

```

      DO 45 I=1,N
      IF ( (I/2) *2 .NE. I ) E=(1,0)
      IF ( (I/2) *2 .EQ. I ) E=(0,1)
      DO 45 J=1,N1+1
      FREQ=DELF*(J-1)-0.5
      U(I,J)=(0.,0.)
      DO 45 K=1,N
      THETA=PI*(N+1-2*K)*FREQ
      U(I,J)=U(I,J)+E*CMPLX(Z(K,I)*COS(THETA),
*      -Z(K,I)*SIN(THETA))
      U1(I,J)=REAL(U(I,J))
45    CONTINUE

```

```

C*****
C*
C*      SECTION FOR LINEAR FOURIER ADDITION OF THE DPSWF
C*      USING SIMPSON'S RULE FOR INTEGRATION
C*
C*****

```

```

      DO 60 I=1,N,2
      COEFF(I)=U1(I,((N1/2)-N2+1))+U1(I,((N1/2)+N2+1))
      DO 50 J=1,N2
50    COEFF(I)=COEFF(I)+4.0*U1(I,((N1/2)-N2)+2.*J))
      DO 55 J=1,N2-1
55    COEFF(I)=COEFF(I)+2.0*U1(I,((N1/2)-N2+1)+2.*J))
      COEFF(I)=COEFF(I)*DELF/3.
60    CONTINUE
      DO 90 I=1,(N+1)/2-1
      DO 65 J=1,N1+1
      FLTER1(I,J)=0.
      DO 65 K=1,2*I+1,2
65    FLTER1(I,J)=FLTER1(I,J)+COEFF(K)*U1(K,J)
      NUMER=FLTER1(I,(N1/2)-N2+1)**2.
*      +FLTER1(I,(N1/2)+N2+1)**2.
      DO 70 J=1,N2
70    NUMER=NUMER+4*FLTER1(I,(N1/2)-N2+2*J)**2.
      DO 75 J=1,N2-1
75    NUMER=NUMER+2*FLTER1(I,(N1/2)-N2+1+2*J)**2.
      DENOM=FLTER1(I,1)**2.+FLTER1(I,N1+1)**2.
      DO 80 J=1,N1/2
80    DENOM=DENOM+4*FLTER1(I,2*J)**2.
      DO 85 J=1,N1/2-1
85    DENOM=DENOM+2*FLTER1(I,1+2*J)**2.
90    ENERGY(I)=NUMER/DENOM

```



```

C*****
C*
C*          SECTION FOR LOGARITHMIC PLOT
C*
C*****

```

```

          DO 95 J=1,N1+1
          U2(J)=20*DLOG10(CDABS(U(1,J))/CDABS(U(1,N1/2+1)))
          IF (U2(J).LT.(-60.)) U2(J)=-60.
95      CONTINUE
          DO 100 I=1,(N+1)/2-1
          DO 100 J=1,N1+1
          FLTER2(I,J)=20*DLOG10(DABS(FLTER1(I,J)
*      /FLTER1(I,N1/2+1)))
          IF (FLTER2(I,J).LT.(-60.)) FLTER2(I,J)=-60.
100     CONTINUE

```

```

C*****
C*
C*          SECTION FOR PLOTTING MAGNITUDE RESPONSE (MAG VS FREQ)
C*
C*****

```

```

          IF (N.GT.9) THEN
          NN=N/10
          III=CHAR(48+NN)
          END IF
          JJJ=CHAR(48+N-NN*10)
          DO 115 I=2,2*((NS+1)/2)+1
          IF (I.GT.9) THEN
          II=I/10
          KKK=CHAR(48+II)
          END IF
          LLL=CHAR(48+I-II*10)
          CALL ASSIGN(I,'E4'//III//JJJ//KKK//LLL//'.DAT')
          WRITE(I,105) N1+1
105     FORMAT(' ',3X,I4)
          DO 115 J=1,N1+1
          FREQ=DELF*(J-1)-0.5
          IF (I.EQ.2) WRITE(I,110) FREQ,U1(1,J)
          IF (I.GE.3.AND.I.LE.(NS+1)/2+1)
*      WRITE(I,110) FREQ,FLTER1(I-2,J)
          IF (I.EQ.(NS+1)/2+2) WRITE(I,110) FREQ,U2(J)
          IF (I.GT.(NS+1)/2+2)
*      WRITE(I,110) FREQ,FLTER2(I-((NS+1)/2+2),J)
110     FORMAT(' ',5X,F8.6,5X,F14.7)
115     CONTINUE

```

```

C*****
C*
C*          OUTPUT SECTION
C*
C*****

```

```

          CALL ASSIGN(1,'E4'//III//JJJ//'.1.DAT')
          DO 145 I=1,N
          WRITE(1,120) I,WR(I),ITS(I),I,COEFF(I)

```



```

120  *  FORMAT(' ', 'LAMBDA(' , I3, ') =', E14.7, 1X, I6,
      *  3X, 'ITERATIONS', 3X, 'COEFF(' , I3, ') =', E14.7)
C    WRITE(1,125)
C 125  FORMAT(' ', 'ASSOCIATED EIGENVECTOR COMPONENTS ARE' /)
C    WRITE(1,130) (Z(L,I), L=1,N)
C 130  FORMAT(' ', 13X, E14.7)
C    WRITE(1,135) SUM1(I)
C 135  FORMAT(' ', 'SUM1=', F9.5)
C    WRITE(1,140) SUM2(I)
C 140  FORMAT(' ', 'SUM2=', F9.5)
145  CONTINUE
      WRITE(1,150) IERR
150  FORMAT(' ', 'THE ERROR CODE IS', I4)
      DO 160 I=1, (N+1)/2-1
      WRITE(1,155) ENERGY(I)
155  *  FORMAT(' ', 'THE ENERGY RATIO OF THE FOURIER FILTER
      *  IS =', F9.6)
160  CONTINUE

```

C\*\*\*\*\*

```

STOP
END

```

```

C*****
C*
C*      ROUTINE TO DETERMINE THE EIGENVALUES AND
C*      EIGENVECTORS OF A SYMMETRIC TRIDIAGONAL MATRIX
C*      (REFERENCE: 'MATRIX EIGENSYSTEM ROUTINES - EISPACK
C*      GUIDE' - SMITH, B.T. et al.)
C*
C*****
C*
C*      LIST OF VARIABLES
C*
C*****
C*
C*      D   = VECTOR OF SIZE N. CONTAINS DIAGONAL ELEMENTS
C*            OF THE SYMMETRIC MATRIX ON INPUT. CONTAINS
C*            EIGENVALUES ON OUTPUT IN ASCENDING ORDER
C*      E   = VECTOR OF SIZE N. CONTAINS N-1 SUBDIAGONAL
C*            ELEMENTS ON INPUT. ROUTINE DESTROYS E ON OUTPUT.
C*            E(1) IS ARBITRARY
C*      ITS = VECTOR OF SIZE N. CONTAINS THE NUMBER OF
C*            ITERATIONS FOR EACH EIGENVALUE
C*
C*****

```

```

      SUBROUTINE TQL2 (NM,N,D,E,Z,IERR,ITS)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ITS(NM)
      REAL*8 D(N),E(N),Z(NM,N),MACHEP

```

```

C*****
C
C      **** MACHEP IS A MACHINE DEPENDENT PARAMETER ****
C      **** SPECIFYING THE RELATIVE PRECISION OF ****
C      **** FLOATING POINT ARITHMETIC ****
      MACHEP=1E-07
      IERR=0
      IF (N.EQ.1) GO TO 80
      DO 5 I=2,N
5      E(I-1)=E(I)
      F=0.0
      B=0.0
      E(N)=0.0
      DO 55 L=1,N
      ITS(L)=0
      H=MACHEP*(DABS(D(L))+DABS(E(L)))
      IF (B.LT.H) B=H
C      **** LOOK FOR SMALL SUB-DIAGONAL ELEMENT ****
      DO 10 M=L,N
      IF (DABS(E(M)).LE.B) GO TO 15
C      **** E(N) IS ALWAYS ZERO, SO THERE IS NO ****
C      **** EXIT THROUGH THE BOTTOM OF THE LOOP ****
10      CONTINUE
15      IF (M.EQ.L) GO TO 50
20      IF (ITS(L).EQ.30) GO TO 75
      ITS(L)=ITS(L)+1
C      **** FORM SHIFT ****
      L1=L+1

```

```

G=D(L)
P=(D(L1)-G)/(2.0*E(L))
R=DSQRT(P*P+1.0)
D(L)=E(L)/(P+DSIGN(R,P))
H=G-D(L)
DO 25 I=L1,N
25 D(I)=D(I)-H
F=F+H
C ***** QL TRANSFORMATION *****
P=D(M)
C=1.0
S=0.0
MML=M-L
DO 45 II=1,MML
I=M-II
G=C*E(I)
H=C*P
IF (DABS(P).LT.DABS(E(I))) GO TO 30
C=E(I)/P
R=DSQRT(C*C+1.0)
E(I+1)=S*P*R
S=C/R
C=1.0/R
GO TO 35
30 C=P/E(I)
R=DSQRT(C*C+1.0)
E(I+1)=S*E(I)*R
S=1.0/R
C=C*S
35 P=C*D(I)-S*G
D(I+1)=H+S*(C*G+S*D(I))
C ***** FORM VECTOR *****
DO 40 K=1,N
H=Z(K,I+1)
Z(K,I+1)=S*Z(K,I)+C*H
Z(K,I)=C*Z(K,I)-S*H
40 CONTINUE
45 CONTINUE
E(L)=S*P
D(L)=C*P
IF (DABS(E(L)).GT.B) GO TO 20
50 D(L)=D(L)+F
55 CONTINUE
C ***** ORDER EIGENVALUES AND EIGENVECTORS *****
DO 70 II=2,N
I=II-1
K=I
P=D(I)
T=ITS(I)
DO 60 J=II,N
IF (D(J).LE.P) GO TO 60
K=J
P=D(J)
T=ITS(J)
60 CONTINUE
IF (K.EQ.I) GO TO 70
D(K)=D(I)

```



```
      ITS(K)=ITS(I)
      D(I)=P
      ITS(I)=T
      DO 65 J=1,N
      P=Z(J,I)
      Z(J,I)=Z(J,K)
      Z(J,K)=P
65    CONTINUE
70    CONTINUE
      GO TO 80
C      **** SET ERROR -- NO CONVERGENCE TO AN ****
C      **** EIGENVALUE AFTER 30 ITERATIONS ****
75    IERR=L
C*****
80    RETURN
      END
```

```

C*****
C*
C*   ROUTINE FOR REDUCING A REAL SYMMETRIC MATRIX TO A
C*   SYMMETRIC TRIDIAGONAL MATRIX ACCUMULATING
C*   ORTHOGONAL TRANSFORMATIONS
C*   (REFERENCE: 'MATRIX EIGENSYSTEM ROUTINES - EISPACK
C*   GUIDE' - SMITH, B.T. et al.)
C*
C*****
C*
C*   LIST OF VARIABLES
C*
C*****
C*
C*   A =  A SYMMETRIC MATRIX OF ORDER N TO BE REDUCED.
C*   ONLY THE FULL LOWER TRIANGLE OF THE MATRIX
C*   NEED BE SUPPLIED
C*   D =  VECTOR OF SIZE N. CONTAINS DIAGONAL ELEMENTS
C*   OF THE SYMMETRIC MATRIX.
C*   E =  VECTOR OF SIZE N. CONTAINS N-1 SUBDIAGONAL
C*   ELEMENTS. E(1) IS ARBITRARY
C*   Z =  ORTHOGONAL TRANSFORMATION MATRIX PRODUCED
C*   IN THE REDUCTION PROCESS
C*
C*****

```

```

SUBROUTINE TRED2 (NM,N,A,D,E,Z)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(NM,N),D(N),E(N),Z(NM,N)

```

```

C*****
C
      DO 5 I=1,N
      DO 5 J=1,I
      Z(I,J)=A(I,J)
5      CONTINUE
      IF (N.EQ.1) GO TO 70
      DO 65 II=2,N
      I=N+2-II
      L=I-1
      H=0.0
      SCALE=0.0
      IF (L.LT.2) GO TO 15
      C      **** SCALE ROW ****
      DO 10 K=1,L
      10      SCALE=SCALE+DABS(Z(I,K))
      IF (SCALE.NE.0.0) GO TO 20
      15      E(I)=Z(I,L)
      GO TO 60
      20      DO 25 K=1,L
      Z(I,K)=Z(I,K)/SCALE
      H=H+Z(I,K)*Z(I,K)
      25      CONTINUE
      F=Z(I,L)
      G=-DSIGN(DSQRT(H),F)
      E(I)=SCALE*G
      H=H-F*G

```

```

      Z(I,L)=F-G
      F=0.0
      DO 45 J=1,L
      Z(J,I)=Z(I,J)/(SCALE*H)
      G=0.0
C     **** FORM ELEMENT OF A*U ****
      DO 30 K=1,J
30    G=G+Z(J,K)*Z(I,K)
      JP1=J+1
      IF(L.LT.JP1)GO TO 40
      DO 35 K=JP1,L
35    G=G+Z(K,J)*Z(I,K)
C     **** FORM ELEMENT OF P ****
40    E(J)=G/H
      F=F+E(J)*Z(I,J)
45    CONTINUE
      HH=F/(H+H)
C     **** FORM REDUCED A ****
      DO 50 J=1,L
      F=Z(I,J)
      G=E(J)-HH*F
      E(J)=G
      DO 50 K=1,J
      Z(J,K)=Z(J,K)-F*E(K)-G*Z(I,K)
50    CONTINUE
      DO 55 K=1,L
55    Z(I,K)=SCALE*Z(I,K)
60    D(I)=H
65    CONTINUE
70    D(1)=0.0
      E(1)=0.0
C     **** ACCUMULATION OF TRANSFORMATION MATRICES ****
      DO 95 I=1,N
      L=I-1
      IF(D(I).EQ.0.0)GO TO 85
      DO 80 J=1,L
      G=0.0
      DO 75 K=1,L
75    G=G+Z(I,K)*Z(K,J)
      DO 80 K=1,L
      Z(K,J)=Z(K,J)-G*Z(K,I)
80    CONTINUE
85    D(I)=Z(I,I)
      Z(I,I)=1.0
      IF(L.LT.1)GO TO 95
      DO 90 J=1,L
      Z(I,J)=0.0
      Z(J,I)=0.0
90    CONTINUE
95    CONTINUE

C*****
      RETURN
      END

```



```

C*****
C*
C*      THIS PROGRAM IS FOR COMPUTING THE FIGURES OF
C*      MERIT OF A FEW WINDOWS
C*
C*****
C*
C*      LIST OF VARIABLES
C*
C*****
C*
C*      N          = ORDER OF THE WINDOW
C*      NS         = NUMBER OF DIFFERENT WINDOWS
C*      NSL        = NUMBER OF SIDELOBES IN WINDOW RESPONSE
C*      N1         = NUMBER OF FREQUENCY POINTS FOR PLOTTING
C*      CG         = COHERENT GAIN
C*      SL         = SCALLOPING LOSS
C*      ENBW       = EQUIVALENT NOISE BANDWIDTH
C*      WL         = WORST CASE PROCESSING LOSS
C*      EMR        = EXPECTED MINIMUM RESPONSE FOR PLOTTING
C*      HSL        = HIGHEST SIDELOBE LEVEL
C*      SLFREQ     = FREQUENCY AT HSL IN BINS
C*      FREQ3DB    = FREQUENCY AT THE THREE dB LEVEL
C*      ACT3DB     = ACTUAL LEVEL AT THAT FREQUENCY
C*      BW3DB      = THREE dB BANDWIDTH IN BINS
C*      FREQ6DB    = FREQUENCY AT THE SIX dB LEVEL
C*      ACT6DB     = ACTUAL LEVEL AT THAT FREQUENCY
C*      BW6DB      = SIX dB BANDWIDTH IN BINS
C*      Z          = PROLATE MATRIX OF EIGENVECTORS
C*      Y(1,N)     = RECTANGLE WINDOW COEFFICIENTS
C*      Y(2,N)     = TRIANGLE WINDOW COEFFICIENTS
C*      Y(3,N)     = 92 dB-BLACKMAN-HARRIS WINDOW COEFFICIENTS
C*      Y(4,N)     = 74 dB-BLACKMAN-HARRIS WINDOW COEFFICIENTS
C*      Y(5,N)     = PROLATE WINDOW WITH W = 0.03
C*      Y(6,N)     = PROLATE WINDOW WITH W = 0.05
C*      Y(7,N)     = PROLATE WINDOW WITH W = 0.06
C*      Y(8,N)     = PROLATE WINDOW WITH W = 0.08
C*      Y(9,N)     = BARCILON-TEMES WINDOW WITH ALPHA = 3.0
C*      Y(10,N)    = BARCILON-TEMES WINDOW WITH ALPHA = 3.5
C*      Y(11,N)    = BARCILON-TEMES WINDOW WITH ALPHA = 4.0
C*      Y(12,N)    = DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 2.5
C*      Y(13,N)    = DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 3.0
C*      Y(14,N)    = DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 3.5
C*      Y(15,N)    = DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 4.0
C*
C*****

```

```

C
C      COMPLEX*16 SL1(15),U(800),V(50)
C      REAL*8 SL(15),Y(15,50),A(4),B(4),W(50),PI,THETA
C      REAL*8 CG(15),DG(15),WL(15),ENBW(15)
C      REAL*8 Z(50,50),T(50),VV(50),FFREQ(200),SFREQ(200)
C      REAL*8 X(800),FREQ(800),U1(800),U2(800)
C      REAL*8 FREQ3DB,FREQ6DB,ACT3DB,ACT6DB,BW3DB,BW6DB
C      COMMON PI
C      DATA A/O.35875,O.48829,O.14128,O.01168/
C      DATA A/O.3635819,O.4891775,O.1365995,O.0106411/
C      DATA A/O.338946,O.481973,O.161054,O.018027/

```

```

DATA B/O.40217,O.49703,O.09892,O.00188/
CALL ASSIGN (1,'E71.DAT')
N=50
NS=15
N1=800
EMR= -120.
PI=2*DASIN(1.DO)
C   RECTANGLE WINDOW
    DO 5 J=1,N
      5   Y(1,J)=1.
C   TRIANGLE WINDOW
    DO 10 J=1,N
      IF (J.LE. (N/2)) Y(2,J)=(2.*J)/N
      IF (J.GE. (N/2)) Y(2,J)=Y(2,N-J)
    10   CONTINUE
C   92 dB-BLACKMAN-HARRIS WINDOW
    DO 15 J=1,N
      THETA=2*PI*(J-1)/N
    15   Y(3,J)=A(1)-A(2)*DCOS(THETA)+A(3)*DCOS(2*THETA)
      *   -A(4)*DCOS(3*THETA)
C   74 dB-BLACKMAN-HARRIS WINDOW
    DO 20 J=1,N
      THETA=2*PI*(J-1)/N
    20   Y(4,J)=B(1)-B(2)*DCOS(THETA)+B(3)*DCOS(2*THETA)
      *   -B(4)*DCOS(3*THETA)
C   PROLATE WINDOW WITH W = 0.03
    CALL DPSS(N,0.03,Z)
    DO 25 J=1,N
      25   Y(5,J)=Z(J,1)
C   PROLATE WINDOW WITH W = 0.05
    CALL DPSS(N,0.05,Z)
    DO 30 J=1,N
      30   Y(6,J)=Z(J,1)
C   PROLATE WINDOW WITH W = 0.06
    CALL DPSS(N,0.06,Z)
    DO 31 J=1,N
      31   Y(7,J)=Z(J,1)
C   PROLATE WINDOW WITH W = 0.08
    CALL DPSS(N,0.08,Z)
    DO 35 J=1,N
      35   Y(8,J)=Z(J,1)
C   BARCILON-TEMES WINDOW WITH ALPHA = 3.0
    CALL BARCITEMES(3.0DO,N,W,T,V,VV)
    DO 40 J=1,N
      40   Y(9,J)=W(J)
C   BARCILON-TEMES WINDOW WITH ALPHA = 3.5
    CALL BARCITEMES(3.5DO,N,W,T,V,VV)
    DO 45 J=1,N
      45   Y(10,J)=W(J)
C   BARCILON-TEMES WINDOW WITH ALPHA = 4.0
    CALL BARCITEMES(4.0DO,N,W,T,V,VV)
    DO 50 J=1,N
      50   Y(11,J)=W(J)
C   DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 2.5
    CALL DOLPHCHEBY(2.5DO,N,W,T,V)
    DO 55 J=1,N
      55   Y(12,J)=W(J)

```



```

CALL DOLPHCHEBY(3.ODO,N,W,T,V)
DO 60 J=1,N
60 Y(13,J)=W(J)
CALL DOLPHCHEBY(3.5DO,N,W,T,V)
DO 65 J=1,N
65 Y(14,J)=W(J)
CALL DOLPHCHEBY(4.ODO,N,W,T,V)
DO 70 J=1,N
70 Y(15,J)=W(J)
DO 150 I=1,NS
IF(I.EQ.1) WRITE(1,*) 'RECTANGLE WINDOW'
IF(I.EQ.2) WRITE(1,*) 'TRIANGLE WINDOW'
IF(I.EQ.3) WRITE(1,*) '92 dB-BLACKMAN-HARRIS WINDOW'
IF(I.EQ.4) WRITE(1,*) '74 dB-BLACKMAN-HARRIS WINDOW'
IF(I.EQ.5) WRITE(1,*) 'PROLATE WINDOW WITH W = 0.03'
IF(I.EQ.6) WRITE(1,*) 'PROLATE WINDOW WITH W = 0.05'
IF(I.EQ.7) WRITE(1,*) 'PROLATE WINDOW WITH W = 0.06'
IF(I.EQ.8) WRITE(1,*) 'PROLATE WINDOW WITH W = 0.08'
IF(I.EQ.9)
* WRITE(1,*) 'BARCILON-TEMES WINDOW WITH ALPHA = 3.0'
IF(I.EQ.10)
* WRITE(1,*) 'BARCILON-TEMES WINDOW WITH ALPHA = 3.5'
IF(I.EQ.11)
* WRITE(1,*) 'BARCILON-TEMES WINDOW WITH ALPHA = 4.0'
IF(I.EQ.12)
* WRITE(1,*) 'DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 2.5'
IF(I.EQ.13)
* WRITE(1,*) 'DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 3.0'
IF(I.EQ.14)
* WRITE(1,*) 'DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 3.5'
IF(I.EQ.15)
* WRITE(1,*) 'DOLPH-CHEBYSHEV WINDOW WITH ALPHA = 4.0'
DO 75 J=1,N
75 CG(I)=CG(I)+Y(I,J)
DO 80 J=1,N
80 DG(I)=DG(I)+Y(I,J)**2.
DO 85 J=1,N
85 SL1(I)=SL1(I)+DCMLX(Y(I,J)*DCOS(PI/N*J),
* -Y(I,J)*DSIN(PI/N*J))
DO 90 J=1,N
90 X(J)=Y(I,J)
CALL RESPONSE(X,N,FREQ,0.5,N1,U)
CALL LPLOT(U,N1,EMR,U1,U2)
CALL SEARCHSLOBE(FREQ,N1,U2,FFREQ,SFREQ,NSL)
K=2
DO 95 J=2,NSL-2
IF(SFREQ(J+1).GT.SFREQ(K)) THEN
HSL=SFREQ(J+1)
K=K+1
GO TO 95
END IF
IF(SFREQ(J+1).LE.SFREQ(K)) HSL=SFREQ(K)
95 CONTINUE
SLFREQ=2*FFREQ(K)*N
CALL SEARCHFREQ(X,N,-3.DO,FREQ3DB,ACT3L,0.0001DO)
CALL SEARCHFREQ(X,N,-6.DO,FREQ6DB,ACT6L,0.0001DO)
BW3DB=2*FREQ3DB*N

```



```

      BW6DB=2*FREQ6DB*N
      SL(I)=CDABS(SL1(I))
      SL(I)=SL(I)/CG(I)
      SL(I)=-20*DLOG10(SL(I))
      ENBW(I)=N*DG(I)/CG(I)**2.
      CG(I)=CG(I)/N
      WL(I)=SL(I)+10*DLOG10(ENBW(I))
      WRITE(1,100)HSL
100   FORMAT('/', ' ', 'HIGHEST SIDELobe LEVEL IS      =',1X,F8.4)
      WRITE(1,105)CG(I)
105   FORMAT(' ', 'THE COHERENT GAIN IS                =',1X,F8.4)
      WRITE(1,110)ENBW(I)
110   FORMAT(' ', 'EQUIVALENT NOISE BANDWIDTH          =',1X,F8.4)
      WRITE(1,115)BW3DB
115   FORMAT(' ', 'THREE dB BANDWIDTH IS (BINS)=',1X,F8.4)
      WRITE(1,120)SL(I)
120   FORMAT(' ', 'THE SCALLOPING LOSS IS              =',1X,F8.4)
      WRITE(1,125)WL(I)
125   FORMAT(' ', 'THE WORSTCASE LOSS IS                =',1X,F8.4)
      WRITE(1,130)BW6DB
130   FORMAT(' ', 'SIX dB BANDWIDTH IS (BINS)          =',1X,F8.4)
      WRITE(1,135)SLFREQ
135   FORMAT(' ', 'FREQUENCY AT SIDELobe (BINS)         =',1X,F8.4)
      WRITE(1,140)ACT3L
140   FORMAT(' ', 'ACTUAL LEVEL AT 3 dB EDGE            =',1X,F8.4)
      WRITE(1,145)ACT6L
145   FORMAT(' ', 'ACTUAL LEVEL AT 6 dB EDGE            =',1X,
      * F8.4//)
150   CONTINUE

```

C\*\*\*\*\*

```

      STOP
      END

```

```

C*****
C*
C*          ROUTINE FOR COMPUTING THE PROLATE WINDOW
C*          COEFFICIENTS
C*
C*****
C*
C*          LIST OF VARIABLES
C*
C*****
C*
C*  A          = MATRIX WHOSE EIGENVECTORS ARE THE DPSS
C*  N          = ORDER OF A
C*  Z(I,L)     = THE N COMPONENTS OF THE Lth EIGENVECTOR
C*  Z(I,1)     = THE WINDOW COEFFICIENTS
C*  WR,WI      = VECTORS CONTAINING EIGENVALUES OF A
C*  NM         = NUMBER OF ROWS IN Z MATRIX DIMENSION
C*  IERR       = ERROR PARAMETER
C*  W          = NORMALISED BANDWIDTH (BETWEEN 0. AND .5)
C*
C*****

```

```

      SUBROUTINE DPSS(N,W,Z)
      INTEGER*4 ITS(50)
      REAL*8 Z(50,50),WR(50),WI(50),A(50,50)
      REAL*8 SUM1(50),SUM2(50),FV1(50),IV1(50)
      REAL*8 Z1

```

```

C*****
C*
C*          THE PARAMETERS ARE ENTERED HERE
C*
C*****

```

```

      NM=N
      PI=ASIN(1.)*2

```

```

C*****
C*
C*          MATRIX ELEMENTS ARE DEFINED HERE
C*
C*****

```

```

      DO 20 I=1,N
      DO 15 J=1,N
      IF (I.EQ.J) GO TO 5
      GO TO 10
      5  A(I,J)=2*W
      GO TO 15
      10 A(I,J)=SIN(PI*2*W*(I-J))/(PI*(I-J))
      15 CONTINUE
      20 CONTINUE

```

```

C*****
C*
C* SECTION FOR COMPUTING EIGENVALUES AND EIGENVECTORS
C*
C*****

```

```

      CALL TRED2 (NM,N,A,WR,FV1,Z)
      CALL TQL2 (NM,N,WR,FV1,Z,IERR,ITS)

```

```

C*****
C*
C* SECTION FOR RENORMALIZING DPSS
C*
C*****

```

```

      DO 21 J=1,N
      Z1=Z(N/2,J)
      DO 21 I=1,N
      Z(I,J)=Z(I,J)/Z1
21    CONTINUE

```

```

C*****
C*
C* SECTION FOR MAKING SUM Z(I,J) > 0, AND
C* SUM (N+1-2*I)*Z(I,J) > 0
C*
C*****

```

```

      DO 40 J=1,N
      SUM1(J)=0.
      SUM2(J)=0.
      DO 25 I=1,N
      SUM1(J)=SUM1(J)+Z(I,J)
25    SUM2(J)=SUM2(J)+(N+1-2*I)*Z(I,J)
      IF (SUM1(J).LT.0.) THEN
      DO 30 I=1,N
30    Z(I,J)=-Z(I,J)
      SUM1(J)=-SUM1(J)
      END IF
      IF (SUM2(J).LT.0.) THEN
      DO 35 I=1,N/2
      TEMP=Z(I,J)
      Z(I,J)=Z(N+1-I,J)
35    Z(N+1-I,J)=TEMP
      SUM2(J)=-SUM2(J)
      END IF
40    CONTINUE

```

```

C*****
      RETURN
      END

```



```

C*****
C*
C*          ROUTINE FOR DOLPH-CHEBYSHEV WINDOW
C*
C*****
C*
C*          LIST OF VARIABLES
C*
C*****
C*      ALPHA = PARAMETER IN THE DESIGN OF THE WINDOW
C*      N      = ORDER OF THE WINDOW
C*      Z      = THE WINDOW COEFFICIENTS
C*
C*****

```

```

      SUBROUTINE DOLPHCHEBY (ALPHA,N,Z,W,Y)
      COMPLEX*16 Y(N)
      REAL*8 A,B,C,D,E,F,G,THETA,Z(N),Z1,ALPHA,PI,W(N)
      COMMON PI
      G=10.**ALPHA
      A=DLOG(G+DSQRT(G**2-1.))
      B=DCOSH(A/N)
      C=DLOG(B+DSQRT(B**2-1.))
      D=DCOSH(N*C)
      DO 5 K=1,N
      THETA=2*PI*K/N
      E=DABS(B*DCOS(PI*K/N))
      IF (E.LT.1.DO) THEN
      F=PI/2-DATAN(E/DSQRT((1.-E**2)))
      W(K)=DCOS(N*F)/D
      END IF
      IF (E.GE.1.DO) THEN
      F=DLOG(E+DSQRT(E**2-1.))
      W(K)=DCOSH(N*F)/D
      END IF
      W(K)=(-1.**K)*W(K)*DCOS(N/2*THETA)
5      CONTINUE
      DO 15 I=1,N
      Y(I)=(0.,0.)
      DO 10 J=1,N
      THETA=2*PI*J*I/N
      Y(I)=Y(I)+(W(J)/N)*DCMPLX(DCOS(THETA),DSIN(THETA))
10     CONTINUE
      Z(I)=CDABS(Y(I))
15     CONTINUE
      Z1=Z(N/2)
      DO 20 J=1,N
      Z(J)=Z(J)/Z1
20     CONTINUE

```

```

C*****

```

```

      RETURN
      END

```

```

C*****
C*
C*          ROUTINE FOR BARCILON-TEMES WINDOW
C*
C*****
C*
C*          LIST OF VARIABLES
C*
C*****
C*
C*      ALPHA = PARAMETER IN THE DESIGN OF THE WINDOW
C*      N      = ORDER OF THE WINDOW
C*      Z      = THE WINDOW COEFFICIENTS
C*
C*****

```

```

      SUBROUTINE BARCITEMES (ALPHA,N,Z,W,Y,X)
      COMPLEX*16 Y(N)
      REAL*8 A,B,C,D,E,F,THETA,Z(N),Z1,ALPHA,PI,X(N),W(N)
      COMMON PI
      A=DSQRT((10.** (2.*ALPHA))-1.)
      B=10.**ALPHA
      C=DLOG(B+DSQRT(B**2-1.))
      D=DCOSH(C/N)
      DO 5 K=1,N
      E=DABS(D*DCOS(PI*K/N))
      IF (E.LT.1.DO) THEN
      F=PI/2-DATAN(E/DSQRT((1.-E**2)))
      X(K)=N*F
      W(K)=(-1**K)*((A*DCOS(X(K))+B*X(K)/C*DSIN(X(K)))/
      * ((C+A*B)*((X(K)/C)**2.+1.)))*DCOS(PI*K)
      END IF
      IF (E.GE.1.DO) THEN
      F=DLOG(E+DSQRT(E**2-1.))
      X(K)=N*F
      W(K)=(-1**K)*((A*DCOSH(X(K))+B*X(K)/C*DSINH(X(K)))/
      * ((C+A*B)*((X(K)/C)**2.+1.)))*DCOS(PI*K)
      END IF
5      CONTINUE
      DO 15 I=1,N
      Y(I)=(0.,0.)
      DO 10 J=1,N
      THETA=2*PI*J*I/N
      Y(I)=Y(I)+(W(J)/N)*DCMLX(DCOS(THETA),DSIN(THETA))
10      CONTINUE
      Z(I)=CDABS(Y(I))
15      CONTINUE
      Z1=Z(N/2)
      DO 20 J=1,N
      Z(J)=Z(J)/Z1
20      CONTINUE

```

```

C*****

```

```

      RETURN
      END

```

```

C*****
C*
C*      ROUTINE FOR COMPUTING THE FREQUENCY RESPONSE      *
C*      OF THE WINDOW                                     *
C*
C*****
C*
C*      LIST OF VARIABLES                                  *
C*
C*****
C*
C*      X      = THE WINDOW COEFFICIENTS                  *
C*      N      = ORDER OF THE WINDOW                     *
C*      N1     = NUMBER OF POINTS FOR FREQUENCY RESPONSE *
C*      FREQ   = THE FREQUENCIES AT WHICH RESPONSE IS COMPUTED *
C*      FREQR  = THE RANGE OF FREQUENCY FOR WHICH RESPONSE *
C*               IS DESIRED (USUALLY -.5 TO .5)           *
C*      U      = FREQUENCY RESPONSE OF THE WINDOW        *
C*
C*****

      SUBROUTINE RESPONSE (X,N,FREQ,FREQR,N1,U)
      COMMON PI
      COMPLEX*16 U(N1+1)
      REAL*8 X(N1+1),FREQ(N1+1),FREQR,THETA

C*****

      DO 5 J=1,N1+1
      FREQ(J)=2.*FREQR/N1*(J-1)-FREQR
      U(J)=(0.,0.)
      DO 5 K=1,N
      THETA=PI*(N+1-2*K)*FREQ(J)
      U(J)=U(J)+DCMPLX(X(K)*DCOS(THETA),-X(K)*DSIN(THETA))
5      CONTINUE

C*****

      RETURN
      END

```



```

C*****
C*
C*          ROUTINE FOR LOGARITHMIC PLOT
C*
C*****
C*
C*          LIST OF VARIABLES
C*
C*****
C*
C*          U   = FREQUENCY RESPONSE OF THE WINDOW
C*          U1  = REAL PART OF U
C*          U2  = RESPONSE IN DECIBELS
C*          N1  = NUMBER OF POINTS FOR FREQUENCY RESPONSE
C*          EMR = EXPECTED MINIMUM RESPONSE FOR PLOTTING
C*
C*****

      SUBROUTINE L PLOT (U,N1,EMR,U1,U2)
      COMPLEX*16 U(N1+1)
      REAL*8 U1(N1+1),U2(N1+1)

C*****

      DO 5 J=1,N1+1
      U1(J)=REAL(U(J))
      U2(J)=20*DLOG10(CDABS(U(J))/CDABS(U(N1/2+1)))
      IF(U2(J).LT.EMR) U2(J)=EMR
5      CONTINUE

C*****

      RETURN
      END

```

```

C*****
C*
C*      ROUTINE FOR SEARCHING THE SIDELobe LEVELS
C*      AND THEIR FREQUENCY LOCATIONS
C*
C*****
C*
C*      LIST OF VARIABLES
C*
C*****
C*      FFREQ  =  FREQUENCY AT WHICH THE SIDELOBES PEAK
C*      SFREQ  =  SIDELobe PEAK LEVELS
C*
C*****

```

```

      SUBROUTINE SEARCHSLOBE (FREQ,N1,U2,FFREQ,SFREQ,I)
      REAL*8 FREQ(N1+1),U2(N1+1),FFREQ(N1/2),SFREQ(N1/2)

```

```

C*****

```

```

      K=1
      DO 5 I=N1/2+1,N1
      IF ( (U2(I-1) .LT. U2(I)) .AND. (U2(I) .GT. U2(I+1)) ) THEN
      FFREQ(K)=FREQ(I)
      SFREQ(K)=U2(I)
      K=K+1
      END IF
5      CONTINUE
      I=K

```

```

C*****

```

```

      RETURN
      END

```

```

C*****
C*
C*   ROUTINE TO SEARCH FOR A FREQUENCY AT A PARTICULAR
C*   DECIBEL LEVEL
C*
C*****
C*   LIST OF VARIABLES
C*****
C*
C*   X           = THE WINDOW COEFFICIENTS
C*   SRCHL       = dB LEVEL BEING SEARCHED
C*   FREQL       = FREQUENCY AT SEARCH dB LEVEL
C*   ACTLEV      = ACTUAL dB AT THAT FREQUENCY
C*   EPSI        = PERMITTED TOLERANCE IN SRCHL
C*   FREQ        = CURRENT FREQUENCY OF SEARCH
C*   DELTAF      = FREQUENCY INCREMENT FOR SEARCHING
C*
C*****

```

```

SUBROUTINE SEARCHFREQ(X,N,SRCHL,FREQL,ACTLEV,EPSI)
COMMON PI
REAL*8 X(N),SRCHL,FREQL,ACTLEV,U2(2)
REAL*8 DIFF,EPSI,THETA,FREQ
COMPLEX*16 U(2),U1

```

```

C*****

```

```

      FREQ=0.DO
      DELTAF=0.1
5     DO 15 I=1,2
      IF (I.EQ.1) FREQ=FREQ
      IF (I.EQ.2) FREQ=FREQ+DELTAF
      U(I)=(0.,0.)
      DO 10 J=1,N
      THETA=PI*(N+1-2.*J)*FREQ
      U(I)=U(I)+DCMPLX(X(J)*DCOS(THETA),-X(J)*DSIN(THETA))
10    CONTINUE
      IF (FREQ.EQ.0.) U1=U(I)
      U2(I)=20*DLOG10(CDABS(U(I))/CDABS(U1))
      DIFF=DABS(U2(I))-DABS(SRCHL)
      IF (DABS(DIFF).LT.EPSI) GO TO 20
15    CONTINUE
      IF ((U2(1).GT.SRCHL).AND.(U2(2).GT.SRCHL)) THEN
      GO TO 5
      END IF
      IF ((U2(1).GT.SRCHL).AND.(U2(2).LT.SRCHL)) THEN
      FREQ=FREQ-DELTAF
      DELTAF=DELTAF/2.
      GO TO 5
      END IF
20    FREQL=FREQ
      ACTLEV=U2(I)

```

```

C*****

```

```

      RETURN
      END

```



```

C*****
C*
C*  THIS PROGRAM IS FOR COMPUTING THE FREQUENCY PLOT  AND
C*  APPLICATION IN THE HARMONIC ANALYSIS EXAMPLE (HARRIS)
C*  OF THE RECTANGLE, BLACKMAN-HARRIS AND PROLATE WINDOWS
C*
C*****
C*
C*                      LIST OF VARIABLES
C*
C*****
C*  N          = ORDER OF THE WINDOW
C*  M          = NUMBER OF FREQUENCY POINTS AT WHICH
C*              SPECTRUM IS COMPUTED. (GENERALLY = N)
C*  N1         = NUMBER OF POINTS FOR FREQUENCY RESPONSE
C*  F(N)       = SINUSOID INPUT PROCESSED BY THE WINDOW
C*  F1,F2      = COMPONENTS IN F(N)
C*  FW(N)      = THE SPECTRUM AT THE OUTPUT OF THE WINDOW
C*  Y(1,N)     = RECTANGULAR WINDOW COEFFICIENTS
C*  Y(2,N)     = BLACKMAN-HARRIS WINDOW COEFFICIENTS
C*  Y(3,N)     = PROLATE WINDOW COEFFICIENTS
C*  MAG(N)     = MAGNITUDE OF FW(N)
C*  U(I,N)     = FREQUENCY RESPONSE OF THE WINDOW
C*  U1(I,N)    = REAL PART OF U
C*  U2(I,N)    = MAGNITUDE OF U
C*
C*****

```

```

C      COMPLEX*16 U(3,100),FW(26)
C      REAL*8 U1(3,100),U2(3,100),Y(3,50),Z1(50,50)
C      REAL*8 F(50),Z(50),MAG(26),A(4)
C      DATA A/.35875,.48829,.14128,.01168/
C      DATA A/O.40217,O.49703,O.09892,O.00188/
C      CALL ASSIGN(1,'HA1.DAT')
C      CALL ASSIGN(2,'HA2.DAT')
C      CALL ASSIGN(3,'HA3.DAT')
C      CALL ASSIGN(4,'HA4.DAT')
C      CALL ASSIGN(5,'HA5.DAT')
C      CALL ASSIGN(6,'HA6.DAT')
C      N=50
C      NS=3
C      N1=80
C      M=N/2+1
C      PI=2*ASIN(1.)
C      DO 30 I=1,NS
C      RECTANGULAR WINDOW
C      IF (I.EQ.1) THEN
C      DO 5 J=1,N
C      Y(I,J)=1.
C      5  Z(J)=Y(I,J)
C      END IF
C      74 dB 4-SAMPLE BLACKMAN-HARRIS WINDOW
C      IF (I.EQ.2) THEN
C      DO 10 J=1,N
C      THETA=2*PI*(J-1)/N

```

```

      Y(I,J)=A(1)-A(2)*COS(THETA)+A(3)*COS(2*THETA)
      -A(4)*COS(3*THETA)
10    Z(J)=Y(I,J)
      END IF
C    PROLATE WINDOW WITH W = 0.06
      IF(I.EQ.3) THEN
      CALL DPSS(N,0.06,Z1)
      DO 11 J=1,N
      Y(I,J)=Z1(J,1)
11    Z(J)=Y(I,J)
      END IF
      CALL HARMONIC (Z,N,MAG,FW,M,F,5.5,10.)
      WRITE(I,15)M
15    FORMAT(' ',I3)
      DO 25 K=1,M
      WRITE(I,20)K-1,MAG(K)
20    FORMAT(' ',I3,5X,F9.5)
25    CONTINUE
30    CONTINUE

C*****
C*
C*      SECTION FOR COMPUTING THE FREQUENCY RESPONSE
C*
C*****

      DO 35 I=1,NS
      DO 35 J=1,N1+1
      FREQ=1./N1*(J-1)-0.5
      U(I,J)=(0.,0.)
      DO 35 K=1,N
      THETA=PI*(N+1-2*K)*FREQ
      U(I,J)=U(I,J)+DCMLPX(Y(I,K)*COS(THETA),
      * -Y(I,K)*SIN(THETA))
      U1(I,J)=REAL(U(I,J))
35    CONTINUE

C*****
C*      SECTION FOR LOGARITHMIC PLOT
C*****

      DO 50 I=1,NS
      WRITE(I+3,40)N1+1
40    FORMAT(' ',I3)
      DO 50 J=1,N1+1
      FREQ=1./N1*(J-1)-0.5
      U2(I,J)=20*DLOG10(CDABS(U(I,J))/CDABS(U(I,N1/2+1)))
      IF(U2(I,J).LT.(-120.))U2(I,J)=-120.
      WRITE(I+3,45)FREQ,U2(I,J)
45    FORMAT(' ',F9.5,3X,F10.5)
50    CONTINUE

C*****

      STOP
      END

```



```

C*****
C*
C*      ROUTINE FOR THE HARMONIC ANALYSIS USING THE
C*      GIVEN WINDOW
C*
C*****
C*
C*      LIST OF VARIABLES
C*
C*****
C*
C*      N      = THE ORDER OF THE WINDOW
C*      M      = THE NUMBER OF FREQUENCY POINTS AT WHICH
C*               SPECTRUM IS COMPUTED. (GENERALLY = N)
C*      F(N)   = THE SINUSOID INPUT THAT IS PROCESSED BY THE
C*               GIVEN WINDOW
C*      F1,F2  = COMPONENTS IN F(N)
C*      FW(N)  = THE SPECTRUM AT THE OUTPUT OF THE WINDOW
C*      W(N)   = THE WINDOW COEFFICIENTS
C*      MAG(N) = THE MAGNITUDE OF FW(N)
C*
C*****

```

```

      SUBROUTINE HARMONIC (W,N,MAG,FW,M,F,F1,F2)
      REAL*8 F(N),W(N),MAG(M),MAG1
      COMPLEX*16 FW(M)
      PI=2*ASIN(1.)
      DO 10 I=1,N
10    F(I)=COS(2*PI*F1*(I-1)/N)+0.01*COS(2*PI*F2*(I-1)/N)
      DO 20 K=1,M
      FREQ=2*PI*(K-1)/N
      FW(K)=(0.,0.)
      DO 20 I=1,N
20    FW(K)=FW(K)+W(I)*F(I)
      *CMPLX(COS(FREQ*(I-1)), -SIN(FREQ*(I-1)))
      DO 30 K=1,M
30    MAG(K)=CDABS(FW(K))
      MAG1=MAG(1)
      DO 40 K=1,M
40    IF (MAG(K) .GE. MAG1) MAG1=MAG(K)
      DO 50 K=1,M
      MAG(K)=20*DLOG10(MAG(K)/MAG1)
      IF (MAG(K) .LT. -70.) MAG(K)=-70.
50    CONTINUE

```

```

C*****

```

```

      RETURN
      END

```



```

C*****
C*
C*          A GENERAL PLOTTING ROUTINE
C*
C*****
C*
C*          VARIABLES ARE INTRODUCED HERE
C*
C*****
C*
C*  IHT      = LOGICAL VARIABLE FOR TITLE OF X AXIS
C*  NHT      = NO. OF LETTERS IN TITLE OF X AXIS
C*  IVT      = LOGICAL VARIABLE FOR TITLE OF Y AXIS
C*  NVT      = NO. OF LETTERS IN TITLE OF Y AXIS
C*  ISMB     = LOGICAL VARIABLE FOR TITLE OF A CURVE
C*  ISMP     = LOGICAL VARIABLE FOR TITLE OF THE FIGURE
C*  IANN     = LOGICAL VARIABLE FOR ANNOTATIONS
C*  ISM      = LOGICAL VARIABLE FOR SYMBOL CODE
C*  ISM1     = MARKER OR NO MARKER IN CURVE
C*  X,Y      = SET OF DATA POINTS
C*  N        = NUMBER OF DATA POINTS
C*  NSAME    = NO. OF CURVES ON SAME FIGURE
C*  NLINES   = NO. OF LINES IN FIGURE TITLE
C*  FILE     = NAME OF DATA FILE
C*  HXL      = LENGTH OF X AXIS
C*  VYL      = LENGTH OF Y AXIS
C*  HOL      = LENGTH OF HORIZONTAL OUTLINE
C*  VOL      = LENGTH OF VERTICAL OUTLINE
C*  FH       = START VALUE ON X AXIS
C*  FV       = START VALUE ON Y AXIS
C*  DH       = INCREMENT ALONG X AXIS
C*  DV       = INCREMENT ALONG Y AXIS
C*  OX       = X COORD. OF PLOTTER ORIGIN
C*  OY       = Y COORD. OF PLOTTER ORIGIN
C*  DX       = X COORD. OF LOWEST LEGEND
C*  DY       = Y COORD. OF LOWEST LEGEND
C*  HLL      = HORIZONTAL LENGTH OF LEGEND BOX
C*  KCYCL    = INCREMENT IN GATHERING DATA FROM THE ARRAYS
C*  LTYPE    = TYPE OF CURVE CONNECTING THE DATA POINTS
C*  NDP      = n FOR LINE WITH MARKERS AT EVERY nth POINT
C*  FCTR     = ENLARGEMENT/REDUCTION FACTOR OF FIGURE
C*  SLANT    = SLANT ANGLE FOR ITALICS
C*  LX       = 1 FOR LOG X AXIS
C*  LY       = 1 FOR LOG Y AXIS
C*  ALSIZE   = SIZE OF AXIS LABELS (THE NUMERALS)
C*  ANSIZE   = SIZE OF LETTERS IN ANNOTATION
C*  ATSIZE   = SIZE OF LETTERS IN AXIS TITLES
C*  CTSIZE   = SIZE OF LETTERS IN CURVE TITLE
C*  FTSIZE   = SIZE OF LETTERS IN FIGURE TITLE
C*  MLSIZE   = SIZE OF MARKER IN LEGEND
C*
C*****

```

```

LOGICAL*4 IVT(50) , IHT(50) , ISMB(50) , ISMP(50) , IANN(50)
REAL MLSIZE
BYTE FILE(15)
DIMENSION X(1002) , Y(1002)

```

```

CALL PLOTS(53,0,-9)
CALL DASHDF(0,0,0,0)
TYPE*, 'WELCOME TO THIS ZETA PLOTTING ROUTINE'
TYPE*, 'ENTER 1, IF YOU WANT TO USE LOG X AXIS'
TYPE*, 'ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, LX
TYPE*, 'ENTER X AXIS LENGTH, START VALUE, INCREMENT'
IF (LX.EQ.1) THEN
TYPE*, '(NOTE: HERE INCREMENT REFERS TO NUMBER OF'
TYPE*, 'LOG CYCLES PER UNIT LENGTH; START VALUE MUST'
TYPE*, 'BE POSITIVE; DO NOT USE CUBIC SPLINE OPTION)'
END IF
ACCEPT*, HXL, FH, DH
TYPE*, 'ENTER 1, IF YOU WANT TO USE LOG Y AXIS'
ACCEPT*, LY
TYPE*, 'ENTER Y AXIS LENGTH, START VALUE, INCREMENT'
IF (LY.EQ.1) THEN
TYPE*, '(NOTE: HERE INCREMENT REFERS TO NUMBER OF'
TYPE*, 'LOG CYCLES PER UNIT LENGTH; START VALUE MUST'
TYPE*, 'BE POSITIVE; DO NOT USE CUBIC SPLINE OPTION)'
END IF
ACCEPT*, VYL, FV, DV
ALSIZE=.100
ANSIZE=.100
ATSIZE=.140
CTSIZE=.125
FTSIZE=.175
MLSIZE=.100
XANGLE=0.
YANGLE=90.
HLL=4.
SLANT=0.
KCYCL=1
TYPE*, 'ENTER 1, TO CHANGE ANY OF THE LETTERING'
TYPE*, 'SIZES. (ENTER ANY OTHER INTEGER OTHERWISE)'
ACCEPT*, K
IF (K.EQ.1) THEN
TYPE*, 'THE FOLLOWING ARE THE DEFAULT SIZES'
TYPE*, '1 - THE AXIS LABEL 0.100'
TYPE*, '2 - THE ANNOTATIONS 0.100'
TYPE*, '3 - THE AXIS TITLE 0.140'
TYPE*, '4 - THE CURVE TITLE 0.125'
TYPE*, '5 - THE FIGURE TITLE 0.175'
TYPE*, '6 - THE MARKER IN LEGEND 0.100'
TYPE*, '
TYPE*, 'HOW MANY OF THEM DO YOU WANT TO CHANGE?'
ACCEPT*, K
DO 4 I=1, K
TYPE*, 'ENTER NUMBER OF ITEM IN THE ABOVE LIST (1-7)'
ACCEPT*, K1
TYPE*, 'ENTER THE SIZE'
IF (K1.EQ.1) ACCEPT*, ALSIZE
IF (K1.EQ.2) ACCEPT*, ANSIZE
IF (K1.EQ.3) ACCEPT*, ATSIZE
IF (K1.EQ.4) ACCEPT*, CTSIZE

```



```

IF (K1.EQ.5) ACCEPT*,FTSIZE
IF (K1.EQ.6) ACCEPT*,MLSIZE
4  CONTINUE
END IF
TYPE*, 'ENTER TITLE OF X AXIS'
ACCEPT5, (IHT(I), I=1,50)
5  FORMAT(50A4)
TYPE*, 'ENTER NO. OF LETTERS IN TITLE'
ACCEPT*, NHT
TYPE*, 'ENTER TITLE OF Y AXIS'
ACCEPT5, (IVT(I), I=1,50)
TYPE*, 'ENTER NO. OF LETTERS IN TITLE'
ACCEPT*, NVT
NHT=-NHT
TYPE*, 'ENTER FACTOR FOR SIZE REDUCTION'
ACCEPT*, F
FCTR=1/F
CALL FACTOR(FCTR)
OX=(8.5/FCTR-HXL)/2
OY=(11/FCTR-VYL)/2
CALL PLOT(OX,OY,-3)
CALL NEWPEN(1)
IF (LX.EQ.1) GO TO 15
CALL AXIS(O.,O.,IHT,NHT,HXL,XANGLE,999.,1.)
IF ((FH.EQ.IFIX(FH)*1.) .AND. (DH.EQ.IFIX(DH)*1.)) K=-1
IF ((FH.NE.IFIX(FH)*1.) .OR. (DH.NE.IFIX(DH)*1.)) K=2
DO 10 I=1, IFIX(HXL)+1
DSX=I-1
FNUM=FH+DH*(I-1)
CALL NUMBER(DSX,-2*ALSIZE,ALSIZE,FNUM,XANGLE,K)
10 CONTINUE
NHT=-NHT
DSX=(HXL-NHT*ATSIZE)/2
CALL SYMBOL(DSX,-4*ALSIZE,ATSIZE,IHT,XANGLE,NHT)
IF (LX.NE.1) GO TO 20
15 CALL LGAXIS(OX,OY,IHT,NHT,HXL,XANGLE,FH,DH)
20 IF (LY.EQ.1) GO TO 30
CALL AXIS(O.,O.,IVT,NVT,VYL,YANGLE,999.,1.)
IF ((FV.EQ.IFIX(FV)*1.) .AND. (DV.EQ.IFIX(DV)*1.)) K=-1
IF ((FV.NE.IFIX(FV)*1.) .OR. (DV.NE.IFIX(DV)*1.)) K=2
DO 25 I=1, IFIX(VYL)+1
DSY=I-1-ALSIZE/2
FNUM=FV+DV*(I-1)
CALL NUMBER(-ALSIZE,DSY,ALSIZE,FNUM,YANGLE,K)
25 CONTINUE
DSY=(VYL-NVT*ATSIZE)/2
CALL SYMBOL(-3*ALSIZE,DSY,ATSIZE,IVT,YANGLE,NVT)
IF (LY.NE.1) GO TO 35
30 CALL LGAXIS(OX,OY,IVT,NVT,VYL,YANGLE,FV,DV)
35 TYPE*, 'ENTER NO. OF CURVES IN THE FIGURE'
ACCEPT*, NSAME
IF (NSAME.EQ.1) THEN
TYPE*, 'ENTER TYPE OF CURVE'
TYPE*, '4 - FOR ONLY MARKERS'
TYPE*, '5 - FOR CUBIC SPLINE WITHOUT MARKER'
TYPE*, '6 - FOR STRAIGHT LINE WITHOUT MARKER'
ACCEPT*, LTYPE

```



```

GO TO 45
END IF
DX=HXL-HLL
DY=VYL
TYPE*, 'LEGEND APPEARS AT RIGHT TOP OF FIGURE'
TYPE*, 'ENTER 1 - IF YOU WANT TO CHANGE THIS'
TYPE*, 'ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K
IF (K.EQ.1) THEN
TYPE*, 'ENTER X,Y COOR. OF LEFT BOTTOM OF LEGEND'
TYPE*, '(WITH RESPECT TO LEFT BOTTOM OF FIGURE)'
ACCEPT*, DX, DY
END IF
TYPE*, 'ENTER 1 IF YOU WANT THE LEGEND IN ITALICS'
TYPE*, 'ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K
IF (K.EQ.1) SLANT=30.0
TYPE*, 'ENTER TYPE OF CURVE'
TYPE*, '1 - FOR CUBIC SPLINE WITH MARKER'
TYPE*, '2 - FOR STRAIGHT LINE WITH MARKER'
TYPE*, '3 - FOR DOT-DASH LINES WITH MARKER'
TYPE*, '4 - FOR ONLY MARKERS'
TYPE*, '5 - FOR CUBIC SPLINE WITHOUT MARKER'
TYPE*, '6 - FOR STRAIGHT LINE WITHOUT MARKER'
TYPE*, '7 - FOR DOT-DASH LINES WITHOUT MARKER'
TYPE*, '8 - FOR MIXED OPTIONS'
ACCEPT*, LTYPE
DSY=DY
DO 85 I=1, NSAME
TYPE40, I
40  FORMAT(25H ENTER TITLE OF CURVE NO., I3)
ACCEPT5, (ISMB(J), J=1, 50)
45  CONTINUE
IF (LTYPE.NE.8) LTYPE1=LTYPE
IF (LTYPE.EQ.8) THEN
TYPE*, 'ENTER TYPE OF CURVE'
TYPE*, '1 - FOR CUBIC SPLINE WITH MARKER'
TYPE*, '2 - FOR STRAIGHT LINE WITH MARKER'
TYPE*, '3 - FOR DOT-DASH LINES WITH MARKER'
TYPE*, '4 - FOR ONLY MARKERS'
TYPE*, '5 - FOR CUBIC SPLINE WITHOUT MARKER'
TYPE*, '6 - FOR STRAIGHT LINE WITHOUT MARKER'
TYPE*, '7 - FOR DOT-DASH LINES WITHOUT MARKER'
ACCEPT*, LTYPE1
END IF
TYPE*, 'ENTER A NUMBER (1-4) TO CHANGE THE PEN'
TYPE*, '1-BLACK; 2-RED; 3-GREEN; 4-BLUE;'
TYPE*, '(NOTE:INITIALLY IT IS PEN NUMBER 1. YOU HAVE'
TYPE*, 'TO SPECIFY THIS ORDER, WHILE SUBMITTING JOB)'
ACCEPT*, K
CALL NEWPEN(K)
IF (LTYPE1.LE.4) THEN
TYPE*, 'ENTER MARKER CODE (0-15)'
ACCEPT*, ISM
ISM1=ISM
TYPE*, 'MARKERS ARE PROVIDED AT EVERY 8th POINT'
TYPE*, 'ENTER 1 - IF YOU WANT TO RETAIN THIS'

```

```

TYPE*, 'ENTER 2 - IF YOU WANT TO CHANGE THIS'
ACCEPT*, K
IF (K.EQ.1) NDP=8
IF (K.EQ.2) THEN
TYPE*, 'ENTER THE MARKER DENSITY'
ACCEPT*, NDP
END IF
END IF
IF ((LTYPE1.EQ.3).OR.(LTYPE1.EQ.7)) THEN
TYPE*, 'ENTER LINE TYPE (1-6)'
ACCEPT*, ISM2
END IF
TYPE*, 'PLEASE ENTER DATA FILE NAME'
ACCEPT50, FILE
FORMAT(15A1)
FILE(15)=0
OPEN(UNIT=1, NAME=FILE, ACCESS='SEQUENTIAL',
* TYPE='OLD')
READ(1, *) N
DO 55 J=1, N
READ(1, *) X(J), Y(J)
55 CONTINUE
CLOSE(UNIT=1)
X(N+1)=FH
Y(N+1)=FV
X(N+2)=DH
Y(N+2)=DV
KCYCL=1
IF (LTYPE1.LE.4) THEN
IF ((N.LT.10).AND.(NDP.NE.1)) THEN
NDP=1
TYPE*, '(MARKERS WILL BE PROVIDED AT EVERY POINT'
TYPE*, 'SINCE NUMBER OF POINTS IS LESS THAN TEN.)'
TYPE*, '
END IF
END IF
IF (LTYPE1.EQ.1) N=-N
IF (LTYPE1.EQ.2) N=N
IF (LTYPE1.EQ.3) THEN
N=-N
KCYCL=ISM2+3
NDP=NDP
ISM1=ISM
IF ((LX.NE.1).AND.(LY.NE.1)) GO TO 60
IF ((LX.EQ.1).AND.(LY.NE.1)) LXY=-1
IF ((LX.EQ.1).AND.(LY.EQ.1)) LXY=0
IF ((LX.NE.1).AND.(LY.EQ.1)) LXY=1
CALL ORIGIN(OX, OY, O)
CALL LGLINE(X, Y, N, KCYCL, NDP, ISM1, LXY)
CALL PLOT(O., O., -3)
GO TO 75
60 CALL LINE(X, Y, N, KCYCL, NDP, ISM1)
GO TO 75
END IF
IF (LTYPE1.EQ.4) NDP=-NDP
IF (LTYPE1.EQ.5) THEN
N=-N

```



```

NDP=0
ISM1=2
END IF
IF (LTYPE1.EQ.6) THEN
N=N
NDP=0
ISM1=2
END IF
IF (LTYPE1.EQ.7) THEN
N=-N
KCYCL=ISM2+3
NDP=0
IF ((LX.NE.1).AND.(LY.NE.1)) GO TO 65
IF ((LX.EQ.1).AND.(LY.NE.1)) LXY=-1
IF ((LX.EQ.1).AND.(LY.EQ.1)) LXY=0
IF ((LX.NE.1).AND.(LY.EQ.1)) LXY=1
CALL ORIGIN(OX,OY,O)
CALL LGLINE(X,Y,N,KCYCL,NDP,ISM1,LXY)
CALL PLOT(O.,O.,-3)
GO TO 75
65 CALL LINE(X,Y,N,KCYCL,NDP,ISM1)
GO TO 75
END IF
IF ((LX.NE.1).AND.(LY.NE.1)) GO TO 70
IF ((LX.EQ.1).AND.(LY.NE.1)) LXY=-1
IF ((LX.EQ.1).AND.(LY.EQ.1)) LXY=0
IF ((LX.NE.1).AND.(LY.EQ.1)) LXY=1
CALL ORIGIN(OX,OY,O)
CALL LGLINE(X,Y,N,KCYCL,NDP,ISM1,LXY)
CALL PLOT(O.,O.,-3)
GO TO 75
70 CALL FLINE(X,Y,N,KCYCL,NDP,ISM1)
75 IF (NSAME.EQ.1) GO TO 90
DSX=DX+0.50
DSY=DSY+2*CTSIZE
IF (LTYPE1.LE.3) THEN
TYPE*, 'ENTER 1, IF YOU WANT A LINE DRAWN WITH MARKER'
TYPE*, 'IN LEGEND. ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K
IF (K.EQ.1) THEN
CALL PLOT(DX+0.25,DSY,3)
CALL PLOT(DX+0.4375,DSY,2)
CALL PLOT(DX+0.5625,DSY,3)
CALL PLOT(DX+0.75,DSY,2)
END IF
END IF
IF ((LTYPE1.EQ.5).OR.(LTYPE1.EQ.6)) THEN
TYPE*, 'ENTER 1, IF YOU WANT A LINE DRAWN IN LEGEND'
TYPE*, 'ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K
IF (K.EQ.1) THEN
DSX=DX+0.25
CALL PLOT(DSX,DSY,3)
CALL PLOT(DSX+0.5,DSY,2)
END IF
GO TO 80
END IF

```



```

      IF (LTYPE1.EQ.7) THEN
        DSX=DX+0.25
        CALL PLOT(DSX,DSY,3)
        CALL PLOT(DSX+0.5,DSY,KCYCL)
        GO TO 80
      END IF
      CALL ITALIC(O.)
      CALL SYMBOL(DSX,DSY,MLSIZE/FCTR,ISM1,XANGLE,-1)
80    DSX=DX+1.25
      CALL ITALIC(SLANT)
      CALL SYMBOL(DSX,DSY-CTSIZE/2,CTSIZE,ISMB,XANGLE,50)
85    CONTINUE
90    CALL NEWPEN(1)
      TYPE*, 'ENTER NUMBER OF LINES IN TITLE'
      ACCEPT*, NLINES
      SLANT=0.
      IF (NLINES.EQ.0) GO TO 110
      TYPE*, 'DO YOU WANT THE TITLE IN ITALICS?'
      TYPE*, 'ENTER 1, IF YES'
      ACCEPT*, K
      IF (K.EQ.1) SLANT=30.0
      DSY=-1.
      DO 105 I=1, NLINES
        TYPE95, I
95    FORMAT(15H ENTER LINE NO., I3, 10H OF TITLE)
        ACCEPT100, (ISMP(J), J=1, 50)
100   FORMAT(50A4)
        TYPE*, 'ENTER NO. OF LETTERS IN TITLE'
        ACCEPT*, K
        DSX=(HXL-K*FTSIZE)/2
        CALL ITALIC(SLANT)
        CALL SYMBOL(DSX,DSY,FTSIZE,ISMP,XANGLE,50)
        DSY=DSY-2*FTSIZE
105   CONTINUE
110   CONTINUE
        CALL PLOT(HXL,O.,3)
        CALL PLOT(HXL,VYL,2)
        CALL PLOT(O.,VYL,2)
        TYPE*, 'ENTER 1, IF YOU WANT CROSS HATCHING'
        TYPE*, 'ENTER ANY OTHER INTEGER OTHERWISE'
        ACCEPT*, K
        IF (K.NE.1) GO TO 170
        IF (LX.EQ.1) GO TO 120
        DO 115 I=1, 100
          OHX1=I
          IF (OHX1.GE.HXL) GO TO 140
          CALL PLOT(OHX1,O.,3)
          OVY1=VYL
          CALL PLOT(OHX1,OVY1,2)
115   CONTINUE
120   IF (FH.LT.1.) FH=FH*10
        IF (FH.GE.1.) GO TO 125
        GO TO 120
125   IF (FH.GT.10.) FH=FH/10
        IF (FH.LE.10.) GO TO 130
        GO TO 125
130   IFH1=IINT(FH)

```

```

OHX1=0.
DO 135 I=1,100
IFH=IFH1+I
J=(I-2+IFH1)/9
IF (IFH.GT.9*J) IFH=IFH-9*J
FH2=FLOAT(IFH)
IF ((I.EQ.1).AND.(FH.GT.FLOAT(IFH1))) FH2=FH+1
OHX1=OHX1+ALOG10(1.*IFH/(FH2-1))/DH
IF (OHX1.GE.HXL) GO TO 140
CALL PLOT(OHX1,0.,3)
OVY1=VYL
CALL PLOT(OHX1,OVY1,2)
135 CONTINUE
140 IF (LY.EQ.1) GO TO 150
DO 145 I=1,100
OVY1=I
IF (OVY1.GE.VYL) GO TO 170
CALL PLOT(0.,OVY1,3)
OHX1=HXL
CALL PLOT(OHX1,OVY1,2)
145 CONTINUE
150 IF (FV.LT.1.) FV=FV*10
IF (FV.GE.1.) GO TO 155
GO TO 150
155 IF (FV.GT.10.) FV=FV/10
IF (FV.LE.10.) GO TO 160
GO TO 155
160 IFV1=IINT(FV)
OVY1=0.
DO 165 I=1,100
IFV=IFV1+I
J=(I-2+IFV1)/9
IF (IFV.GT.9*J) IFV=IFV-9*J
FV2=FLOAT(IFV)
IF ((I.EQ.1).AND.(FV.GT.FLOAT(IFV1))) FV2=FV+1
OVY1=OVY1+ALOG10(1.*IFV/(FV2-1))/DV
IF (OVY1.GE.VYL) GO TO 170
CALL PLOT(0.,OVY1,3)
OHX1=HXL
CALL PLOT(OHX1,OVY1,2)
165 CONTINUE
170 CONTINUE
IF (NSAME.NE.1) THEN
TYPE*, 'ENTER 1, IF YOU WANT THE LEGEND BOXED.'
TYPE*, 'ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K
IF (K.EQ.1) THEN
TYPE*, 'THE LEGEND BOX HAS A LENGTH OF 4 UNITS.'
TYPE*, 'ENTER 1 - IF YOU WANT TO CHANGE THIS.'
TYPE*, 'ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K
IF (K.EQ.1) THEN
TYPE*, 'ENTER LENGTH OF LEGEND BOX'
ACCEPT*, HLL
END IF
CALL PLOT(DX+HLL,DY,3)
CALL PLOT(DX+HLL,DY+3*CTSIZE+NSAME*2*CTSIZE,2)

```



```

CALL PLOT(DX,DY+3*CTSIZE+NSAME*2*CTSIZE,2)
CALL PLOT(DX,DY,2)
IF (DY.NE.VYL) CALL PLOT(DX+HLL,DY,2)
IF ((DY.EQ.VYL).AND.((DX+HLL).GT.HXL)) THEN
CALL PLOT(HXL,VYL,3)
CALL PLOT(DX+HLL,VYL,2)
END IF
IF ((DY.EQ.VYL).AND.(DX.LT.O.)) THEN
CALL PLOT(O.,DY,2)
END IF
END IF
END IF
TYPE*, 'ENTER 1, IF YOU WANT EXTRA ANNOTATIONS, IN'
TYPE*, 'THE PLOT. ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K1
IF (K1.EQ.1) THEN
TYPE*, 'ENTER NUMBER OF ANNOTATIONS'
ACCEPT*, K2
DO 190 I=1,K2
TYPE175,I
175 FORMAT(36H ENTER X,Y COORDINATES OF ANNOTATION,I3)
TYPE*, '(WITH RESPECT TO THE ORIGIN OF THE PLOT)'
ACCEPT*, XANN,YANN
CALL PLOT(XANN,YANN,3)
ANSIZE1=ANSIZE
TYPE*, 'ENTER 1, IF YOU WANT TO CHANGE SIZE OF THIS'
TYPE*, 'ANNOTATION. ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K3
IF (K3.EQ.1) THEN
TYPE*, 'ENTER SIZE OF ANNOTATION'
ACCEPT*, ANSIZE1
END IF
SLANT=0.
TYPE*, 'ENTER 1 IF YOU WANT ANNOTATION IN ITALICS?'
TYPE*, 'ENTER ANY OTHER INTEGER OTHERWISE'
ACCEPT*, K4
IF (K4.EQ.1) SLANT=30.0
TYPE*, 'ARE THERE SPECIAL CHARACTERS IN ANNOTATION?'
TYPE*, 'ENTER 1, IF YES. ENTER 0, OTHERWISE'
ACCEPT*, K5
IF (K5.EQ.1) THEN
TYPE*, 'SPECIAL CHARACTERS (SUCH AS GREEK LETTERS)'
TYPE*, 'ARE ENTERED ONE AT A TIME, BY SPECIFYING'
TYPE*, 'THEIR ASCII CODE. (CHECK PLOTTER MANUAL FOR'
TYPE*, 'LIST OF AVAILABLE CHARACTERS)'
TYPE*, '
TYPE*, 'ENTER NUMBER OF CHARACTERS IN THE STRING'
TYPE*, '(INCLUDE EVEN, NON-SPECIAL CHARACTERS)'
ACCEPT*, K6
DO 185 J=1,K6
TYPE180,J
180 FORMAT(34H ENTER ASCII CODE OF CHARACTER NO.,I3)
ACCEPT*, K7
CALL ITALIC(SLANT)
CALL SYMBOL(999.,999.,ANSIZE1,K7,XANGLE,O)
185 CONTINUE
END IF

```



```

IF (K5.EQ.0) THEN
TYPE*, 'ENTER NUMBER OF CHARACTERS IN THE STRING'
ACCEPT*, K6
TYPE*, 'ENTER THE CHARACTER STRING'
ACCEPT5, (IANN(J), J=1, K6)
CALL ITALIC (SLANT)
CALL SYMBOL (999., 999., ANSIZE1, IANN, XANGLE, K6)
END IF
190 CONTINUE
END IF
TYPE*, 'DO YOU WANT THE A-4 SIZE OUTLINE?'
TYPE*, 'ENTER 0 FOR NO OUTLINE'
TYPE*, 'ENTER 1 FOR 11 BY 8.5 OUTLINE'
TYPE*, 'ENTER 2 FOR 8.5 BY 11 OUTLINE'
ACCEPT*, K
HOL=11.
VOL=8.5
IF (K.EQ.0) GO TO 200
IF (K.EQ.1) GO TO 195
IF (K.EQ.2) THEN
HOL=8.5
VOL=11.
END IF
195 X1=(-HOL/FCTR+HXL)/2
Y1=(-VOL/FCTR+VYL)/2
TYPE*, 'THE A4-SIZE OUTLINE CENTERS THE AXES'
TYPE*, 'ALONG THE CORRESPONDING BOUNDARY LINES.'
TYPE*, 'ENTER 1, IF YOU WANT TO CHANGE THIS'
ACCEPT*, K
IF (K.EQ.1) THEN
TYPE*, 'ENTER X,Y COORDINATES OF LEFT BOTTOM'
TYPE*, 'OF THE A4-SIZE OUTLINE'
ACCEPT*, X1, Y1
END IF
CALL PLOT (X1, Y1, -3)
CALL PLOT (HOL/FCTR, 0, 2)
CALL PLOT (HOL/FCTR, VOL/FCTR, 2)
CALL PLOT (0, VOL/FCTR, 2)
CALL PLOT (0, 0, 2)
200 CALL PLOT (8.5/FCTR-OX, 11/FCTR-OY, 999)

```

C\*\*\*\*\*

```

STOP
END

```



